Creating your own dataset

Install the Transformers, Datasets, and Evaluate libraries to run this notebook.

```
[ ] !pip install datasets evaluate transformers[sentencepiece]
    !apt install git-lfs
```

You will need to setup git, adapt your email and name in the following cell.

```
[ ] !git config --global user.email "you@example.com"
    !git config --global user.name "Your Name"
```

You will also need to be logged in to the Hugging Face Hub. Execute the following and enter your credentials.

```
[ ] from huggingface_hub import notebook_login

    notebook_login()
```

```
[ ] !pip install requests
```

```
[ ] import requests

    url = "https://api.github.com/repos/huggingface/datasets/issues?page=1&per_page=1"
    response = requests.get(url)
```

```
[ ] response.status_code

    200
```

```
[ ] response.json()

    [{'url': 'https://api.github.com/repos/huggingface/datasets/issues/2792',
      'repository_url': 'https://api.github.com/repos/huggingface/datasets',
      'labels_url': 'https://api.github.com/repos/huggingface/datasets/issues/2792/labels{/name}',
      'comments_url': 'https://api.github.com/repos/huggingface/datasets/issues/2792/comments',
      'events_url': 'https://api.github.com/repos/huggingface/datasets/issues/2792/events',
      'html_url': 'https://github.com/huggingface/datasets/pull/2792',
```

```
 'id': 968650274,
 'node_id': 'MDExOlB1bGxSZXF1ZXN0NzEwNzUyMjc0',
 'number': 2792,
 'title': 'Update GooAQ',
 'user': {'login': 'bhavitvyamalik',
  'id': 19718818,
  'node_id': 'MDQ6VXNlcjE5NzE4ODE4',
  'avatar_url': 'https://avatars.githubusercontent.com/u/19718818?v=4',
  'gravatar_id': '',
  'url': 'https://api.github.com/users/bhavitvyamalik',
  'html_url': 'https://github.com/bhavitvyamalik',
  'followers_url': 'https://api.github.com/users/bhavitvyamalik/followers',
  'following_url': 'https://api.github.com/users/bhavitvyamalik/following{/other_user}',
  'gists_url': 'https://api.github.com/users/bhavitvyamalik/gists{/gist_id}',
  'starred_url': 'https://api.github.com/users/bhavitvyamalik/starred{/owner}{/repo}',
  'subscriptions_url': 'https://api.github.com/users/bhavitvyamalik/subscriptions',
  'organizations_url': 'https://api.github.com/users/bhavitvyamalik/orgs',
  'repos_url': 'https://api.github.com/users/bhavitvyamalik/repos',
  'events_url': 'https://api.github.com/users/bhavitvyamalik/events{/privacy}',
  'received_events_url': 'https://api.github.com/users/bhavitvyamalik/received_events',
  'type': 'User',
  'site_admin': False},
 'labels': [],
 'state': 'open',
 'locked': False,
 'assignee': None,
 'assignees': [],
 'milestone': None,
 'comments': 1,
 'created_at': '2021-08-12T11:40:18Z',
 'updated_at': '2021-08-12T12:31:17Z',
 'closed_at': None,
 'author_association': 'CONTRIBUTOR',
 'active_lock_reason': None,
 'pull_request': {'url': 'https://api.github.com/repos/huggingface/datasets/pulls/2792',
  'html_url': 'https://github.com/huggingface/datasets/pull/2792',
  'diff_url': 'https://github.com/huggingface/datasets/pull/2792.diff',
  'patch_url': 'https://github.com/huggingface/datasets/pull/2792.patch'},
 'body': '[GooAQ](https://github.com/allenai/gooaq) dataset was recently updated after splits were added for the same. This PR contains new updated GooAQ wi
and updated README as well.',
 'performed_via_github_app': None}]
```


[ ]
```
GITHUB_TOKEN = xxx  # Copy your GitHub token here
headers = {"Authorization": f"token {GITHUB_TOKEN}"}
```

[ ]
```
import time
import math
from pathlib import Path
import pandas as pd
from tqdm.notebook import tqdm


def fetch_issues(
    owner="huggingface",
    repo="datasets",
```

```
    batch = []
    all_issues = []
    per_page = 100  # Number of issues to return per page
    num_pages = math.ceil(num_issues / per_page)
    base_url = "https://api.github.com/repos"

    for page in tqdm(range(num_pages)):
        # Query with state=all to get both open and closed issues
        query = f"issues?page={page}&per_page={per_page}&state=all"
        issues = requests.get(f"{base_url}/{owner}/{repo}/{query}", headers=headers)
        batch.extend(issues.json())

        if len(batch) > rate_limit and len(all_issues) < num_issues:
            all_issues.extend(batch)
            batch = []  # Flush batch for next time period
            print(f"Reached GitHub rate limit. Sleeping for one hour ...")
            time.sleep(60 * 60 + 1)

    all_issues.extend(batch)
    df = pd.DataFrame.from_records(all_issues)
    df.to_json(f"{issues_path}/{repo}-issues.jsonl", orient="records", lines=True)
    print(
        f"Downloaded all the issues for {repo}! Dataset stored at {issues_path}/{repo}-issues.jsonl"
    )
```

```
[ ] # Depending on your internet connection, this can take several minutes to run...
    fetch_issues()
```

```
[ ] issues_dataset = load_dataset("json", data_files="datasets-issues.jsonl", split="train")
    issues_dataset
```

```
Dataset({
    features: ['url', 'repository_url', 'labels_url', 'comments_url', 'events_url', 'html_url', 'id', 'node_id', 'number', 'title', 'user', 'labels', 'state'
'assignees', 'milestone', 'comments', 'created_at', 'updated_at', 'closed_at', 'author_association', 'active_lock_reason', 'pull_request', 'body', 'timeline_
'performed_via_github_app'],
    num_rows: 3019
})
```

```
[ ] sample = issues_dataset.shuffle(seed=666).select(range(3))

    # Print out the URL and pull request entries
    for url, pr in zip(sample["html_url"], sample["pull_request"]):
        print(f">> URL: {url}")
        print(f">> Pull request: {pr}\n")
```

```
>> URL: https://github.com/huggingface/datasets/pull/850
>> Pull request: {'url': 'https://api.github.com/repos/huggingface/datasets/pulls/850', 'html_url': 'https://github.com/huggingface/datasets/pull/850', 'diff_
'https://github.com/huggingface/datasets/pull/850.diff', 'patch_url': 'https://github.com/huggingface/datasets/pull/850.patch'}

>> URL: https://github.com/huggingface/datasets/issues/2773
>> Pull request: None
```

```
>> URL: https://github.com/huggingface/datasets/pull/783
>> Pull request: {'url': 'https://api.github.com/repos/huggingface/datasets/pulls/783', 'html_url': 'https://github.com/huggingface/datasets/pull/783', 'diff_
'https://github.com/huggingface/datasets/pull/783.diff', 'patch_url': 'https://github.com/huggingface/datasets/pull/783.patch'}
```

[ ]
```
issues_dataset = issues_dataset.map(
    lambda x: {"is_pull_request": False if x["pull_request"] is None else True}
)
```

[ ]
```
issue_number = 2792
url = f"https://api.github.com/repos/huggingface/datasets/issues/{issue_number}/comments"
response = requests.get(url, headers=headers)
response.json()
```


```
[{'url': 'https://api.github.com/repos/huggingface/datasets/issues/comments/897594128',
  'html_url': 'https://github.com/huggingface/datasets/pull/2792#issuecomment-897594128',
  'issue_url': 'https://api.github.com/repos/huggingface/datasets/issues/2792',
  'id': 897594128,
  'node_id': 'IC_kwDODunzps41gDMQ',
  'user': {'login': 'bhavitvyamalik',
   'id': 19718818,
   'node_id': 'MDQ6VXNlcjE5NzE4ODE4',
   'avatar_url': 'https://avatars.githubusercontent.com/u/19718818?v=4',
   'gravatar_id': '',
   'url': 'https://api.github.com/users/bhavitvyamalik',
   'html_url': 'https://github.com/bhavitvyamalik',
   'followers_url': 'https://api.github.com/users/bhavitvyamalik/followers',
   'following_url': 'https://api.github.com/users/bhavitvyamalik/following{/other_user}',
   'gists_url': 'https://api.github.com/users/bhavitvyamalik/gists{/gist_id}',
   'starred_url': 'https://api.github.com/users/bhavitvyamalik/starred{/owner}{/repo}',
   'subscriptions_url': 'https://api.github.com/users/bhavitvyamalik/subscriptions',
   'organizations_url': 'https://api.github.com/users/bhavitvyamalik/orgs',
   'repos_url': 'https://api.github.com/users/bhavitvyamalik/repos',
   'events_url': 'https://api.github.com/users/bhavitvyamalik/events{/privacy}',
   'received_events_url': 'https://api.github.com/users/bhavitvyamalik/received_events',
   'type': 'User',
   'site_admin': False},
  'created_at': '2021-08-12T12:21:52Z',
  'updated_at': '2021-08-12T12:31:17Z',
  'author_association': 'CONTRIBUTOR',
  'body': "@albertvillanova my tests are failing here:\r\n```\r\ndataset_name = 'gooaq'\r\n\r\n    def test_load_dataset(self, dataset_name):\r\n        conf
self.dataset_tester.load_all_configs(dataset_name, is_local=True)[:1]\r\n>       self.dataset_tester.check_load_dataset(dataset_name, configs, is_local=True,
use_local_dummy_data=True)\r\n\r\ntests/test_dataset_common.py:234: \r\n_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
\r\ntests/test_dataset_common.py:187: in check_load_dataset\r\n    self.parent.assertTrue(len(dataset[split]) > 0)\r\nE   AssertionError: False is not true\r
loading dataset on local machine it works fine. Any suggestions on how can I avoid this error?",
  'performed_via_github_app': None}]
```


[ ]
```
def get_comments(issue_number):
    url = f"https://api.github.com/repos/huggingface/datasets/issues/{issue_number}/comments"
    response = requests.get(url, headers=headers)
    return [r["body"] for r in response.json()]


# Test our function works as expected
```

```
get_comments(2792)
```

```
["@albertvillanova my tests are failing here:\r\n```\r\ndataset_name = 'gooaq'\r\n\r\n    def test_load_dataset(self, dataset_name):\r\n        configs =
self.dataset_tester.load_all_configs(dataset_name, is_local=True)[:1]\r\n>       self.dataset_tester.check_load_dataset(dataset_name, configs, is_local=True,
use_local_dummy_data=True)\r\n\r\ntests/test_dataset_common.py:234: \r\n_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
\r\ntests/test_dataset_common.py:187: in check_load_dataset\r\n    self.parent.assertTrue(len(dataset[split]) > 0)\r\nE   AssertionError: False is not true\r
loading dataset on local machine it works fine. Any suggestions on how can I avoid this error?"]
```

[ ]
```
# Depending on your internet connection, this can take a few minutes...
issues_with_comments_dataset = issues_dataset.map(
    lambda x: {"comments": get_comments(x["number"])}
)
```

[ ]
```
from huggingface_hub import notebook_login

notebook_login()
```

[ ]
```
issues_with_comments_dataset.push_to_hub("github-issues")
```

[ ]
```
remote_dataset = load_dataset("lewtun/github-issues", split="train")
remote_dataset
```

```
Dataset({
    features: ['url', 'repository_url', 'labels_url', 'comments_url', 'events_url', 'html_url', 'id', 'node_id', 'number', 'title', 'user', 'labels', 'state'
'assignees', 'milestone', 'comments', 'created_at', 'updated_at', 'closed_at', 'author_association', 'active_lock_reason', 'pull_request', 'body', 'performed
'is_pull_request'],
    num_rows: 2855
})
```