ShowNotImplemented

November 8, 2023

[4]:
```
from src.unitxt.blocks import (
    LoadHF
)
os.chdir('/home/dafnapension/unitxt')
loader=LoadHF(path='GEM/xlsum', name='english')
loader.process()
```

```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
Cell In[4], line 6
      4 os.chdir('/home/dafnapension/unitxt')
      5 loader=LoadHF(path='GEM/xlsum', name='english')
----> 6 loader.process()

File ~/unitxt/src/unitxt/loaders.py:42, in LoadHF.process(self)
     41 def process(self):
---> 42     dataset = hf_load_dataset(
     43␣
  ↪        self.path, name=self.name, data_dir=self.data_dir, data_files=self.da a_files, str
     44     )
     46     return MultiStream.from_iterables(dataset)

File ~/.virtualenvs/myvirtualenv/lib/python3.10/site-packages/datasets/load.py:
  ↪2146, in load_dataset(path, name, data_dir, data_files, split, cache_dir,␣
  ↪features, download_config, download_mode, verification_mode,␣
  ↪ignore_verifications, keep_in_memory, save_infos, revision, token,␣
  ↪use_auth_token, task, streaming, num_proc, storage_options, **config_kwargs)
   2144 # Return iterable dataset in case of streaming
   2145 if streaming:
-> 2146     return builder_instance.as_streaming_dataset(split=split)
   2148 # Some datasets are already processed on the HF google storage
   2149 # Don't try downloading from Google storage for the packaged datasets a␣
  ↪text, json, csv or pandas
   2150 try_from_hf_gcs = path not in _PACKAGED_DATASETS_MODULES

File ~/.virtualenvs/myvirtualenv/lib/python3.10/site-packages/datasets/builder.
  ↪py:1329, in DatasetBuilder.as_streaming_dataset(self, split, base_path)
   1322 dl_manager = StreamingDownloadManager(
```

```
   1323     base_path=base_path or self.base_path,
   1324     download_config=DownloadConfig(token=self.token,␣
 ↪storage_options=self.storage_options),
   1325     dataset_name=self.dataset_name,
   1326     data_dir=self.config.data_dir,
   1327 )
   1328 self._check_manual_download(dl_manager)
-> 1329 splits_generators = {sg.name: sg for sg in␣
 ↪self._split_generators(dl_manager)}
   1330 # By default, return all splits
   1331 if split is None:

File ~/.cache/huggingface/modules/datasets_modules/datasets/GEM--xlsum/
 ↪eb0c1bd988fe61962620fe73722ebc91e0fd5729b4c8acbf3e3b3c50f8b22a96/xlsum.py:131␣
 ↪in Xlsum._split_generators(self, dl_manager)
    128 lang = str(self.config.name)
    129 url = _URL.format(lang, self.VERSION.version_str[:-2])
--> 131 data_dir = dl_manager.download_and_extract(url)
    132 return [
    133     datasets.SplitGenerator(
    134         name=datasets.Split.TRAIN,
   (…)
    153     ),
    154 ]

File ~/.virtualenvs/myvirtualenv/lib/python3.10/site-packages/datasets/download
 ↪streaming_download_manager.py:1063, in StreamingDownloadManager.
 ↪download_and_extract(self, url_or_urls)
   1045 def download_and_extract(self, url_or_urls):
   1046     """Prepare given `url_or_urls` for streaming (add extraction␣
 ↪protocol).
   1047
   1048     This is the lazy version of `DownloadManager.download_and_extract`␣
 ↪for streaming.
   (…)
   1061         url(s): (`str` or `list` or `dict`), URL(s) to stream data from␣
 ↪matching the given input `url_or_urls`.
   1062     """
-> 1063     return self.extract(self.download(url_or_urls))

File ~/.virtualenvs/myvirtualenv/lib/python3.10/site-packages/datasets/download
 ↪streaming_download_manager.py:1015, in StreamingDownloadManager.extract(self,␣
 ↪url_or_urls)
    996 def extract(self, url_or_urls):
    997     """Add extraction protocol for given url(s) for streaming.
    998
    999     This is the lazy version of `DownloadManager.extract` for streaming
   (…)
```

```
   1013     ```
   1014     """
-> 1015     urlpaths = map_nested(self._extract, url_or_urls, map_tuple=True)
   1016     return urlpaths

File ~/.virtualenvs/myvirtualenv/lib/python3.10/site-packages/datasets/utils/
 ↪py_utils.py:456, in map_nested(function, data_struct, dict_only, map_list,␣
 ↪map_tuple, map_numpy, num_proc, parallel_min_length, types, disable_tqdm, des:)
    454 # Singleton
    455 if not isinstance(data_struct, dict) and not isinstance(data_struct,␣
 ↪types):
--> 456     return function(data_struct)
    458 disable_tqdm = disable_tqdm or not logging.is_progress_bar_enabled()
    459 iterable = list(data_struct.values()) if isinstance(data_struct, dict)␣
 ↪else data_struct

File ~/.virtualenvs/myvirtualenv/lib/python3.10/site-packages/datasets/download
 ↪streaming_download_manager.py:1025, in StreamingDownloadManager._extract(self␣
 ↪urlpath)
   1023 extension = _get_path_extension(path)
   1024 if extension in ["tgz", "tar"] or path.endswith((".tar.gz", ".tar.bz2",
 ↪".tar.xz")):
-> 1025     raise NotImplementedError(
   1026         f"Extraction protocol for TAR archives like '{urlpath}' is not␣
 ↪implemented in streaming mode. "
   1027         f"Please use `dl_manager.iter_archive` instead.\n\n"
   1028         f"Example usage:\n\n"
   1029         f"\turl = dl_manager.download(url)\n"
   1030         f"\ttar_archive_iterator = dl_manager.iter_archive(url)\n\n"
   1031         f"\tfor filename, file in tar_archive_iterator:\n"
   1032         f"\t\t…"
   1033     )
   1034 if protocol is None:
   1035     # no extraction
   1036     return urlpath

NotImplementedError: Extraction protocol for TAR archives like 'https://
 ↪huggingface.co/datasets/GEM/xlsum/resolve/main/data/english_XLSum_v2.0.tar.
 ↪bz2' is not implemented in streaming mode. Please use `dl_manager.
 ↪iter_archive` instead.

Example usage:

        url = dl_manager.download(url)
        tar_archive_iterator = dl_manager.iter_archive(url)

        for filename, file in tar_archive_iterator:
                …
```

[ ]: