Process Dataset with Built-in Recipes

Data-Juicer provides lots of built-in data recipes, which are demos or effect-proven. In this notebook, we will start processing a dataet with a built-in data recipe to learn how to use Data-Juicer quickly.

We will start to get familiar with the data processing with a built-in demo recipe.

Start to Process Dataset

To process data using Data-Juicer, you can run `process_data.py` tool with your config as the argument when in the root directory of Data-Juicer, or run the `dj-process` command (an executable wrapper of `process_data.py` tool) with your config anywhere after installing Data-Juicer.

```
# assuming you are in the data-juicer root directory already.

# run the process_data tool in the root dir of Data-Juicer
python tools/process_data.py --config configs/demo/process.yaml

# or run the dj-process command
dj-process --config configs/demo/process.yaml
```

The configs/demo/process.yaml here is the given data_recipes.

```
# Process config example for dataset

# global parameters
project_name: 'demo-process'
dataset_path: './demos/data/demo-dataset.jsonl'  # path to your
dataset directory or file
np: 4  # number of subprocess to process your dataset

export_path: './outputs/demo-process/demo-processed.jsonl'

# process schedule
# a list of several process operators with their arguments
process:
  - language_id_score_filter:
      lang: 'zh'
      min_score: 0.8
```

In this config, we specify the project name, input and output dataset path, number of processors to process the dataset in parallel. In the OP list, which is specified by the `process` schedule, we only add a signle OP `language_id_score_filter`, which will identify the language of the texts in each sample and give a confidence score as stats. We set the taget language label to "zh" and minimum score threshold to 0.8, which means we only keep the samples whose texts are in Chinese with a confidence score larger than or equal to 0.8.

You can run this demo after specifying the correct config file path.

First, we need to go to the root dir of Data-Juicer. You can replace this path with the correct path on your machine.

In [1]:
```
cd ../
```

```
/root/projects/data-juicer
```

```
/usr/local/python310/lib/python3.10/site-packages/IPython/core/magics/osm.
py:417: UserWarning: This is now an optional IPython functionality, settin
g dhist requires you to install the `pickleshare` library.
  self.shell.db['dhist'] = compress_dhist(dhist)[-100:]
```

In [1]:
```
# Then you can run this command in your CLI
!dj-process --config configs/demo/process.yaml
```

```
2024-08-07 17:04:17 | INFO     | data_juicer.config.config:618 - Back up t
he input config file [/root/projects/data-juicer/configs/demo/process.yam
l] into the work_dir [/root/projects/data-juicer/outputs/demo-process]
2024-08-07 17:04:17 | INFO     | data_juicer.config.config:640 - Configura
tion table:
╒═══════════════════════════╤═══════════════════════════════════════════════
═══════════════════════════╕
│ key                       │ values
│
╞═══════════════════════════╪═══════════════════════════════════════════════
═══════════════════════════╡
│ config                    │ [Path_fr(configs/demo/process.yaml, cwd=/root/
projects/data-juicer)]    │
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ hpo_config                │ None
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ data_probe_algo           │ 'uniform'
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ data_probe_ratio          │ 1.0
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ project_name              │ 'demo-process'
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ executor_type             │ 'default'
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ dataset_path              │ '/root/projects/data-juicer/demos/data/demo-da
taset.jsonl'              │
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ export_path               │ '/root/projects/data-juicer/outputs/demo-proce
ss/demo-processed.jsonl'  │
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ export_shard_size         │ 0
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ export_in_parallel        │ False
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ keep_stats_in_res_ds      │ False
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ keep_hashes_in_res_ds     │ False
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ np                        │ 4
```

text_keys	'text'
image_key	'images'
image_special_token	'<__dj__image>'
audio_key	'audios'
audio_special_token	'<__dj__audio>'
video_key	'videos'
video_special_token	'<__dj__video>'
eoc_special_token	'<|__dj__eoc|>'
suffixes	[]
use_cache	True
ds_cache_dir	'/root/.cache/huggingface/datasets'
cache_compress	None
use_checkpoint	False
temp_dir	None
open_tracer	False

```
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| op_list_to_trace              | []
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| trace_num                     | 10
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| op_fusion                     | False
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| process                       | [{'language_id_score_filter': {'accelerator':
None,                         |
|                               |                                      'audio_key': 'a
udios',                       |
|                               |                                      'cpu_required':
1,                          |
|                               |                                      'image_key': 'i
mages',                       |
|                               |                                      'lang': 'zh',
|
|                               |                                      'mem_required':
0,                          |
|                               |                                      'min_score': 0.
8,                            |
|                               |                                      'num_proc': 4,
|
|                               |                                      'stats_export_p
ath': None,                   |
|                               |                                      'text_key': 'te
xt',                          |
|                               |                                      'video_key': 'v
ideos'}}]                     |
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| percentiles                   | []
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| export_original_dataset       | False
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| save_stats_in_one_file        | False
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| ray_address                   | 'auto'
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| debug                         | False
|
|-------------------------------|-------------------------------------------------------------
-------------------------------|
| work_dir                      | '/root/projects/data-juicer/outputs/demo-proce
```

```
ss'                          │
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
│ timestamp                │ '20240807170416'
│
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
│ dataset_dir              │ '/root/projects/data-juicer/demos/data'
│
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
│ add_suffix               │ False
│
╘══════════════════════════╧══════════════════════════════════════════════════════════════
══════════════════════════╛
2024-08-07 17:04:17 | INFO     | data_juicer.core.executor:47 - Using cach
e compression method: [None]
2024-08-07 17:04:17 | INFO     | data_juicer.core.executor:52 - Setting up
data formatter...
2024-08-07 17:04:17 | INFO     | data_juicer.core.executor:74 - Preparing
exporter...
2024-08-07 17:04:17 | INFO     | data_juicer.core.executor:151 - Loading d
ataset from data formatter...
Setting num_proc from 4 back to 1 for the jsonl split to disable multiproc
essing as it only contains one shard.
Generating jsonl split: 6 examples [00:00, 1306.30 examples/s]
2024-08-07 17:04:18 | INFO     | data_juicer.format.formatter:185 - Unifyi
ng the input dataset formats...
2024-08-07 17:04:18 | INFO     | data_juicer.format.formatter:200 - There
are 6 sample(s) in the original dataset.
Filter (num_proc=4): 100%|##########| 6/6 [00:00<00:00, 57.80 examples/s]
2024-08-07 17:04:18 | INFO     | data_juicer.format.formatter:214 - 6 samp
les left after filtering empty text.
2024-08-07 17:04:18 | INFO     | data_juicer.format.mixture_formatter:137
- sampled 6 from 6
2024-08-07 17:04:18 | INFO     | data_juicer.format.mixture_formatter:143
- There are 6 in final dataset
2024-08-07 17:04:18 | INFO     | data_juicer.core.executor:157 - Preparing
process operators...
2024-08-07 17:04:18 | INFO     | data_juicer.utils.model_utils:103 - Loadi
ng fasttext language identification model...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
2024-08-07 17:04:18 | INFO     | data_juicer.utils.model_utils:74 - Model
[/root/.cache/data_juicer/models/lid.176.bin] not found . Downloading...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
2024-08-07 17:04:28 | INFO     | data_juicer.core.executor:164 - Processin
g data...
Adding new column for stats (num_proc=4): 100%|##########| 6/6 [00:00<00:0
0, 66.88 examples/s]
language_id_score_filter_compute_stats (num_proc=4):   0%|          | 0/6
[00:00<?, ? examples/s]2024-08-07 17:04:28 | INFO     | data_juicer.utils.
model_utils:103 - Loading fasttext language identification model...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
2024-08-07 17:04:28 | INFO     | data_juicer.utils.model_utils:103 - Loadi
ng fasttext language identification model...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
```

```
2024-08-07 17:04:28 | INFO     | data_juicer.utils.model_utils:103 - Loadi
ng fasttext language identification model...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
2024-08-07 17:04:28 | INFO     | data_juicer.utils.model_utils:103 - Loadi
ng fasttext language identification model...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
language_id_score_filter_compute_stats (num_proc=4): 100%|##########| 6/6
[00:00<00:00, 35.64 examples/s]
language_id_score_filter_process (num_proc=4): 100%|##########| 6/6 [00:00
<00:00, 68.64 examples/s]
2024-08-07 17:04:28 | INFO     | data_juicer.core.data:193 - OP [language_
id_score_filter] Done in 0.478s. Left 2 samples.
2024-08-07 17:04:28 | INFO     | data_juicer.core.executor:171 - All OPs a
re done in 0.478s.
2024-08-07 17:04:28 | INFO     | data_juicer.core.executor:174 - Exporting
dataset to disk...
2024-08-07 17:04:28 | INFO     | data_juicer.core.exporter:111 - Exporting
computed stats into a single file...
Creating json from Arrow format: 100%|##########| 1/1 [00:00<00:00, 206.82
ba/s]
2024-08-07 17:04:28 | INFO     | data_juicer.core.exporter:140 - Export da
taset into a single file...
Creating json from Arrow format: 100%|##########| 1/1 [00:00<00:00, 1115.8
0ba/s]
```

As we can see in the output log, Data-Juicer:

1. backs up the config file into the work directory, then print the config table in detail.
2. starts to prepare OPs in the process list and exporter for result dataset storage, preprocess the input dataset to a unified format, and load the models used in these OPs.
3. starts to process the dataset in the order of process list and report the processing information of each OP (e.g. time cost, number of left samples).
4. exports the result dataset to the disk.

We will check the whole procedure from the code perspective to better understand what Data-Juicer does during data processing.

The `process_data.py` tool and `dj-processw` tool will call the `Executor.run()` method to process the data. The `Executor` class is a entry class which integrates the whole processing procedure. For now, Data-Juicer supports 2 types of `Executor`: one is for standalone computer and the other is for distributed processing. The latter one will be introduced in the later notebooks. Here we focus on the default standalone `Executor`.

```python
# tools/process_data.py
from loguru import logger
from data_juicer.config import init_configs
from data_juicer.core import Executor

@logger.catch(reraise=True)
def main():
```

```
    cfg = init_configs()
    if cfg.executor_type == 'default':
        executor = Executor(cfg)
    elif cfg.executor_type == 'ray':
        from data_juicer.core.ray_executor import RayExecutor
        executor = RayExecutor(cfg)
    executor.run()


if __name__ == '__main__':
    main()
```

We can also run this part of code in a Python file with specified config file.

In [2]:
```
# we init the corresponding config
from loguru import logger
from data_juicer.config import init_configs
cfg = init_configs(['--config', 'configs/demo/process.yaml'])

from data_juicer.core import Executor
executor = Executor(cfg)
dataset = executor.run()
```

```
/usr/local/python310/lib/python3.10/site-packages/tqdm/auto.py:21: TqdmWar
ning: IProgress not found. Please update jupyter and ipywidgets. See http
s://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
2024-08-07 17:27:49 | INFO     | data_juicer.config.config:618 - Back up t
he input config file [/root/projects/data-juicer/configs/demo/process.yam
l] into the work_dir [/root/projects/data-juicer/outputs/demo-process]
2024-08-07 17:27:49 | INFO     | data_juicer.config.config:640 - Configura
tion table:
╒═══════════════════════════╤═══════════════════════════════════════════════
═══════════════════════════╕
│ key                       │ values
│
╞═══════════════════════════╪═══════════════════════════════════════════════
═══════════════════════════╡
│ config                    │ [Path_fr(configs/demo/process.yaml, cwd=/root/
projects/data-juicer)]    │
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ hpo_config                │ None
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ data_probe_algo           │ 'uniform'
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ data_probe_ratio          │ 1.0
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ project_name              │ 'demo-process'
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ executor_type             │ 'default'
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ dataset_path              │ '/root/projects/data-juicer/demos/data/demo-da
taset.jsonl'              │
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ export_path               │ '/root/projects/data-juicer/outputs/demo-proce
ss/demo-processed.jsonl'  │
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ export_shard_size         │ 0
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ export_in_parallel        │ False
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ keep_stats_in_res_ds      │ False
│
├───────────────────────────┼───────────────────────────────────────────────
──────────────────────────┤
│ keep_hashes_in_res_ds     │ False
```

np	4
text_keys	'text'
image_key	'images'
image_special_token	'<__dj__image>'
audio_key	'audios'
audio_special_token	'<__dj__audio>'
video_key	'videos'
video_special_token	'<__dj__video>'
eoc_special_token	'<|__dj__eoc|>'
suffixes	[]
use_cache	True
ds_cache_dir	'/root/.cache/huggingface/datasets'
cache_compress	None
use_checkpoint	False
temp_dir	None

```
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ open_tracer              │ False
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ op_list_to_trace         │ []
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ trace_num                │ 10
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ op_fusion                │ False
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ process                  │ [{'language_id_score_filter': {'accelerator':
None,                     │
│                          │                                'audio_key': 'a
udios',                   │
│                          │                                'cpu_required':
1,                      │
│                          │                                'image_key': 'i
mages',                   │
│                          │                                'lang': 'zh',
│
│                          │                                'mem_required':
0,                      │
│                          │                                'min_score': 0.
8,                        │
│                          │                                'num_proc': 4,
│
│                          │                                'stats_export_p
ath': None,               │
│                          │                                'text_key': 'te
xt',                      │
│                          │                                'video_key': 'v
ideos'}}]                 │
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ percentiles              │ []
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ export_original_dataset  │ False
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ save_stats_in_one_file   │ False
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ ray_address              │ 'auto'
│
├──────────────────────────┼──────────────────────────────────────────────
──────────────────────────┤
│ debug                    │ False
```

```
|
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
| work_dir                 | '/root/projects/data-juicer/outputs/demo-proce
ss'                       |
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
| timestamp                | '20240807172748'
|
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
| dataset_dir              | '/root/projects/data-juicer/demos/data'
|
├──────────────────────────┼──────────────────────────────────────────────────────────────
──────────────────────────┤
| add_suffix               | False
|
╘══════════════════════════╧══════════════════════════════════════════════════════════════
══════════════════════════╛
2024-08-07 17:27:49 | INFO     | data_juicer.core.executor:47 - Using cach
e compression method: [None]
2024-08-07 17:27:49 | INFO     | data_juicer.core.executor:52 - Setting up
data formatter...
2024-08-07 17:27:49 | INFO     | data_juicer.core.executor:74 - Preparing
exporter...
2024-08-07 17:27:49 | INFO     | data_juicer.core.executor:151 - Loading d
ataset from data formatter...
2024-08-07 17:27:50 | INFO     | data_juicer.format.formatter:185 - Unifyi
ng the input dataset formats...
2024-08-07 17:27:50 | INFO     | data_juicer.format.formatter:200 - There
are 6 sample(s) in the original dataset.
2024-08-07 17:27:50 | INFO     | data_juicer.format.formatter:214 - 6 samp
les left after filtering empty text.
2024-08-07 17:27:50 | INFO     | data_juicer.format.mixture_formatter:137
- sampled 6 from 6
2024-08-07 17:27:50 | INFO     | data_juicer.format.mixture_formatter:143
- There are 6 in final dataset
2024-08-07 17:27:50 | INFO     | data_juicer.core.executor:157 - Preparing
process operators...
2024-08-07 17:27:50 | INFO     | data_juicer.utils.model_utils:103 - Loadi
ng fasttext language identification model...
Warning : `load_model` does not return WordVectorModel or SupervisedModel
any more, but a `FastText` object which is very similar.
2024-08-07 17:27:50 | INFO     | data_juicer.core.executor:164 - Processin
g data...
2024-08-07 17:27:50 | INFO     | data_juicer.core.data:193 - OP [language_
id_score_filter] Done in 0.031s. Left 2 samples.
2024-08-07 17:27:50 | INFO     | data_juicer.core.executor:171 - All OPs a
re done in 0.031s.
2024-08-07 17:27:50 | INFO     | data_juicer.core.executor:174 - Exporting
dataset to disk...
2024-08-07 17:27:50 | INFO     | data_juicer.core.exporter:111 - Exporting
computed stats into a single file...
Creating json from Arrow format: 100%|##########| 1/1 [00:00<00:00, 219.86
ba/s]
2024-08-07 17:27:50 | INFO     | data_juicer.core.exporter:140 - Export da
taset into a single file...
Creating json from Arrow format: 100%|##########| 1/1 [00:00<00:00, 860.19
ba/s]
```

As we can see, the log contains less contents than the run above and this OP is much faster than before. That's because we already process with this recipe before and generate caches for the processing procedure, and this second run only need to load the cache instead of processing again when the recipe is the same.

What `Executor.run()` does during processing

Now, we explain the key `Executor.run()` method in executor.py step by step.

First the method loads and format the input dataset.

We can load dataset from checkpoints in previous runs, or load dataset from the dataset file using data formatter.

```
class Executor:
    ...
    def run(self, load_data_np=None):
        ...
        # 1. format data
        if self.cfg.use_checkpoint and
self.ckpt_manager.ckpt_available:
            logger.info('Loading dataset from checkpoint...')
            dataset = self.ckpt_manager.load_ckpt()
        else:
            logger.info('Loading dataset from data formatter...')
            if load_data_np is None:
                load_data_np = self.cfg.np
            dataset = self.formatter.load_dataset(load_data_np,
self.cfg)
        ...
    ...
```

You can run the code below to check the loaded dataset here interactively.

In [3]:
```
loaded_dataset = executor.formatter.load_dataset(cfg.np, cfg)
loaded_dataset
```

```
2024-08-07 17:35:33 | INFO     | data_juicer.format.formatter:185 - Unifyi
ng the input dataset formats...
2024-08-07 17:35:33 | INFO     | data_juicer.format.formatter:200 - There
are 6 sample(s) in the original dataset.
2024-08-07 17:35:33 | INFO     | data_juicer.format.formatter:214 - 6 samp
les left after filtering empty text.
2024-08-07 17:35:33 | INFO     | data_juicer.format.mixture_formatter:137
- sampled 6 from 6
2024-08-07 17:35:33 | INFO     | data_juicer.format.mixture_formatter:143
- There are 6 in final dataset
2024-08-07 17:43:11 | INFO     | data_juicer.core.data:193 - OP [language_
id_score_filter] Done in 0.031s. Left 2 samples.
2024-08-07 17:43:29 | INFO     | data_juicer.core.data:193 - OP [language_
id_score_filter] Done in 0.029s. Left 2 samples.
/usr/local/python310/lib/python3.10/site-packages/IPython/core/magics/osm.
py:417: UserWarning: This is now an optional IPython functionality, settin
g dhist requires you to install the `pickleshare` library.
  self.shell.db['dhist'] = compress_dhist(dhist)[-100:]
/root/projects/data-juicer
```

Out[3]:
```
Dataset({
    features: ['text', 'meta'],
    num_rows: 6
})
```

Then the method load the OPs according to the process list from the given config file.

```
class Executor:
    ...
    def run(self, load_data_np = None)
        ...
        # 2. extract processes
        logger.info('Preparing process operators...')
        self.process_list, self.ops = load_ops(self.cfg.process,

self.cfg.op_fusion)
        ...
    ...
```

You can run the code below to check the process_list and OPs loaded from the input config file.

In [4]:
```
process_list = executor.process_list
process_list
```

Out[4]:
```
[{'language_id_score_filter': {'lang': 'zh',
   'min_score': 0.8,
   'text_key': 'text',
   'image_key': 'images',
   'audio_key': 'audios',
   'video_key': 'videos',
   'accelerator': None,
   'num_proc': 4,
   'cpu_required': 1,
   'mem_required': 0,
   'stats_export_path': None}}]
```

In [5]:
```
ops = executor.ops
ops
```

Out[5]:
```
[<data_juicer.ops.filter.language_id_score_filter.LanguageIDScoreFilter
at 0x7f079f4e1030>]
```

According to the loaded self.process_list and self.ops, the method runs these OPs to the dataset.

```
class Executor:
    ...
    def run():
        ...
        # 3. data process
        # - If tracer is open, trace each op after it's processed
        # - If checkpoint is open, clean the cache files after
each process
        logger.info('Processing data...')
        tstart = time()
        dataset = dataset.process(self.ops,
                                  exporter=self.exporter,
                                  checkpointer=self.ckpt_manager,
                                  tracer=self.tracer)
        tend = time()
        logger.info(f'All OPs are done in {tend - tstart:.3f}s.')
    ...
```

Here we use the `dataset.process` method on the whole OP list directly. You can run the code below to check this step.

In [8]:
```
res_dataset = loaded_dataset.process(executor.ops)
```

After all the ops are processed, the method dumps the result dataset to the given export path.

```
class Executor:
    ...
    def run():
        ...
        # 4. data export
        logger.info('Exporting dataset to disk...')
        self.exporter.export(dataset)
        ...
    ...
```

You can check the process dataset in the export path in configs/demo/process.yaml

```
export_path: './outputs/demo-process/demo-processed.jsonl'
```

In [9]:
```
# the exported dataset after run
res_dataset
```

Out[9]:
```
Dataset({
    features: ['text', 'meta', '__dj__stats__'],
    num_rows: 2
})
```

Conclusion

In this notebook, we learn how to process our dataset using a built-in data recipe in Data-Juicer, and understand the details during processing in `Executor.run()` method step by step.