Data-Juicer Intermediate Dataset Format for Multimodal Datasets

Due to large format diversity among different multimodal datasets and works, Data-Juicer propose a novel intermediate text-based interleaved data format for multimodal dataset, which is based on chunk-wise formats such MMC4 dataset.

In the Data-Juicer format, a multimodal sample or document is based on a text, which consists of several text chunks. Each chunk is a semantic unit, and all the multimodal information in a chunk should talk about the same thing and be aligned with each other.

Here is a multimodal sample example in Data-Juicer format below.

- It includes 4 chunks split by the special token `<|__dj__eoc|>`.
- In addition to texts, there are 3 other modalities: images, audios, videos. They are stored on the disk and their paths are listed in the corresponding first-level fields in the sample.
- Other modalities are represented as special tokens in the text (e.g. image -- `<__dj__image>`). The special tokens of each modality correspond to the paths in the order of appearance. (e.g. the two image tokens in the third chunk are images of antarctica_map and europe_map respectively)
- There could be multiple types of modalities and multiple modality special tokens in a single chunk, and they are semantically aligned with each other and text in this chunk. The position of special tokens can be random in a chunk. (In general, they are usually before or after the text.)
- For multimodal samples, unlike text-only samples, the computed stats for other modalities could be a list of stats for the list of multimodal data (e.g. image_widths in this sample).

```
{
  "text": "<__dj__image> Antarctica is Earth's southernmost and least-populated continent. <|__dj__eoc|> "
          "<__dj__video> <__dj__audio> Situated almost entirely south of the Antarctic Circle and surrounded by the "
          "Southern Ocean (also known as the Antarctic Ocean), it contains the geographic South Pole. <|__dj__eoc|> "
          "Antarctica is the fifth-largest continent, being about 40% larger than Europe, "
          "and has an area of 14,200,000 km2 (5,500,000 sq mi). <__dj__image> <__dj__image> <|__dj__eoc|> "
          "Most of Antarctica is covered by the Antarctic ice sheet, "
          "with an average thickness of 1.9 km (1.2 mi). <|__dj__eoc|>",
  "images": [
    "path/to/the/image/of/antarctica_snowfield",
    "path/to/the/image/of/antarctica_map",
    "path/to/the/image/of/europe_map"
  ],
  "audios": [
    "path/to/the/audio/of/sound_of_waves_in_Antarctic_Ocean"
  ],
  "videos": [
    "path/to/the/video/of/remote_sensing_view_of_antarctica"
  ],
  "meta": {
    "src": "customized",
    "version": "0.1",
    "author": "xxx"
  },
  "stats": {
    "lang": "en",
    "image_widths": [224, 336, 512],
    ...
  }
}
```

Dataset Format Conversion Tools

According to the intermediate format, Data-Juicer provides several dataset format conversion tools for some popular multimodal works, such as LLaVA, MMC4, WavCaps, Video-ChatGPT, and so on.

These tools consist of two types:

- Other format to Data-Juicer format: These tools are in `source_format_to_data_juicer_format` directory. They help to convert datasets in other formats to target datasets in Data-Juicer format.
- Data-Juicer format to other format: These tools are in `data_juicer_format_to_target_format` directory. They help to convert datasets in Data-Juicer formats to target datasets in target format.

Here we take LLaVA-like dataset as an example to show you how to convert them to Data-Juicer intermediate format and convert back.

LLaVA-like Dataset for Example

Below is a original sample in LLaVA format. As we can see, each sample consists of 3 level-1 fields: "id", "image", and "conversations". The conversion in field "conversations" could be single or multiple turns. We can convert it an interleaved image-text sample format in Data-Juicer intermediate format. Let's begin!

First, we write this example sample to a file.

In [1]:

```python
import json
original_llava_data = [
  {
    "id": "000000033471",
    "image": "coco/train2017/000000033471.jpg",
    "conversations": [
      {
        "from": "human",
        "value": "<image>\nWhat are the colors of the bus in the image?"
      },
      {
        "from": "gpt",
        "value": "The bus in the image is white and red."
      },
      {
        "from": "human",
        "value": "What feature can be seen on the back of the bus?"
      },
      {
        "from": "gpt",
        "value": "The back of the bus features an advertisement."
      },
      {
        "from": "human",
        "value": "Is the bus driving down the street or pulled off to the
      },
      {
        "from": "gpt",
        "value": "The bus is driving down the street, which is crowded wi
      }
    ]
  }
]

with open('llava.json', 'w') as file:
    file.write(json.dumps(original_llava_data, indent=2))
```

Now, we can convert it to Data-Juicer Format with `llava_to_dj.py` tool in conversion tools. For conversation with multiple turns, we convert it into the same text chunk and only put the image in the first turn. And for each turn, we also add the speaker roles before each sentence. Different speakers in different turns are separated by a newline character '\n'.

In [2]:

```
# you can replace the tool path with the correct path on your environment
!python ../tools/multimodal/source_format_to_data_juicer_format/llava_to_
dj_data = json.load(open('dj.jsonl', 'r'))

print(json.dumps(dj_data, indent=2))
```

```
2024-08-06 20:06:14.032 | INFO     | __main__:main:161 - Loading original
LLaVA dataset.
2024-08-06 20:06:14.032 | INFO     | __main__:main:163 - Load [1] samples.
100%|██████████████████████████████████████████| 1/1 [00:00<00:00, 19239.9
3it/s]
2024-08-06 20:06:14.034 | INFO     | __main__:main:287 - Store the target
dataset into [dj.jsonl].
{
  "id": "000000033471",
  "text": "[[human]]: <image>\nWhat are the colors of the bus in the imag
e?\n[[gpt]]: The bus in the image is white and red.\n[[human]]: What featu
re can be seen on the back of the bus?\n[[gpt]]: The back of the bus featu
res an advertisement.\n[[human]]: Is the bus driving down the street or pu
lled off to the side?\n[[gpt]]: The bus is driving down the street, which
is crowded with people and other vehicles. <|__dj__eoc|>",
  "images": [
    "coco/train2017/000000033471.jpg"
  ]
}
```

After processing with Data-Juicer, it can be converted back into LLaVA format, and used in the LLava training process.

In [3]:

```
# you can replace the tool path with the correct path on your environment
!python ../tools/multimodal/data_juicer_format_to_target_format/dj_to_lla
dj_data = json.load(open('llava.json', 'r'))

print(json.dumps(dj_data, indent=2))
```

```
2024-08-06 20:06:46.638 | INFO     | __main__:main:149 - Start to convert.
1it [00:00, 10230.01it/s]
2024-08-06 20:06:46.640 | INFO     | __main__:main:235 - Start to write th
e converted dataset to [llava.json]...
[
  {
    "id": "000000033471",
    "conversations": [
      {
        "from": "human",
        "value": "<image>\nWhat are the colors of the bus in the image?"
      },
      {
        "from": "gpt",
        "value": "The bus in the image is white and red."
      },
      {
        "from": "human",
        "value": "What feature can be seen on the back of the bus?"
      },
      {
        "from": "gpt",
        "value": "The back of the bus features an advertisement."
      },
      {
        "from": "human",
        "value": "Is the bus driving down the street or pulled off to the
side?"
      },
      {
        "from": "gpt",
        "value": "The bus is driving down the street, which is crowded wit
h people and other vehicles."
      }
    ],
    "image": "coco/train2017/000000033471.jpg"
  }
]
```

Finally, you can clean up the generated temperary files.

In [4]:

```
!rm llava.json
!rm dj.jsonl
```

Conclusion

In this notebook, we dive into the details of Data-Juicer intermediate multimodal dataset format and know how to convert datasets in other format to this Data-Juicer format and vice versa using a LLaVA-like example dataset.