Current `jsut` support PyTorch.

As interface, PyTorch's `Dataset` and PyTorch-Lightning's `DataModule` are provided.

JSUT corpus is speech corpus, so we provide `waveform` dataset and `spectrogram` dataset for both interfaces.

- PyTorch
    - (pure PyTorch) dataset
        - waveform: `JSUT_wave`
        - spectrogram: `JSUT_spec`
    - PyTorch-Lightning
        - waveform: `JSUT_wave_DataModule`
        - spectrogram: `JSUT_spec_DataModule`

Original paper

paper arxiv.1711.00354

```
@misc{1711.00354,

Author = {Ryosuke Sonobe and Shinnosuke Takamichi and Hiroshi Saruwatari},

Title = {JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis},

Year = {2017},

Eprint = {arXiv:1711.00354},

}
```

Dependency Notes

PyTorch version

PyTorch version: PyTorch v1.6 is working (We checked with v1.6.0).

Current `jsut` support PyTorch.

As interface, PyTorch's `Dataset` and PyTorch-Lightning's `DataModule` are provided.

JSUT corpus is speech corpus, so we provide `waveform` dataset and `spectrogram` dataset for both interfaces.

- PyTorch
    - (pure PyTorch) dataset
        - waveform: `JSUT_wave`
        - spectrogram: `JSUT_spec`
    - PyTorch-Lightning
        - waveform: `JSUT_wave_DataModule`
        - spectrogram: `JSUT_spec_DataModule`

Original paper

paper arxiv.1711.00354

```
@misc{1711.00354,
Author = {Ryosuke Sonobe and Shinnosuke Takamichi and Hiroshi Saruwatari},
Title = {JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis},

Year = {2017},

Eprint = {arXiv:1711.00354},

}
```

Dependency Notes

PyTorch version

PyTorch version: PyTorch v1.6 is working (We checked with v1.6.0).

For dependency resolution, we do **NOT** explicitly specify the compatible versions.
PyTorch have several distributions for various environment (e.g. compatible CUDA version.)
Unfortunately it make dependency version management complicated for dependency management system.
In our case, the system `poetry` cannot handle cuda variant string
(e.g. `torch>=1.6.0` cannot accept `1.6.0+cu101`.)
In order to resolve this problem, we use `torch==*`, it is equal to no version specification.
`Setup.py` could resolve this problem (e.g. `torchaudio`'s `setup.py`), but we will not bet our effort to this hacky method.