muscima Documentation

Release 0.3.0

Author

Oct 16, 2018

Contents

1 Requirements 3

2 Installation 5

3 First steps 7

4 Contents 9

5 Indices and tables 11

The `muscima` package implements tools for easier manipulation of the MUSCIMA++ dataset. Download the dataset here:

https://ufal.mff.cuni.cz/muscima/download

A description of the dataset is on the project's homepage:

https://ufal.mff.cuni.cz/muscima

And more thoroughly in an arXiv.org publication:

https://arxiv.org/pdf/1703.04824.pdf

This pacakge is licensed under the MIT license (see `LICENSE.txt` file). The package author is Jan Hajič jr. You can contact him at:

```
hajicj@ufal.mff.cuni.cz
```

Questions and comments are welcome! This package is also hosted on github, so if you find a bug, submit an issue (or a pull request!) there:

https://github.com/hajicj/muscima

CHAPTER 1

Requirements

Python 3.5, otherwise nothing beyond the `requirements.txt` file: `lxml` and `numpy`. If you want to apply pitch inference, you should also get `music21`.

CHAPTER 2

Installation

If you have `pip`, just run:

```
pip install muscima
```

If you don't have `pip`, then you should get it. Or use the Anaconda distribution.

CHAPTER 3

First steps

Let’s first download the dataset:

```
curl https://ufal.mff.cuni.cz/~hajicj/2017/docs/MUSCIMA_0.9.zip > MUSCIMA++_0.9.zip
unzip MUSCIMA++_0.9.zip
cd MUSCIMA++_0.9
```

Take a look at the dataset’s `README.md` file first. You can also read it online:

https://ufal.mff.cuni.cz/muscima

Please make sure you understand the license requirements – the data is licensed as CC-BY-NC-SA 4.0, and because it is built over a previous dataset, there are *two* attributions required.

Next, we fire up `ipython` (or just the plain `python` console, but definitely check out ipython if you don’t use it!) and parse the data:

```
ipython
>>> import os
>>> from muscima.io import parse_cropobject_list
>>> cropobject_fnames = [os.path.join('data', 'cropobjects', f) for f in os.listdir(
↪'data/cropobjects')]
>>> docs = [parse_cropobject_list(f) for f in cropobject_fnames]
>>> len(docs)
140
```

In `docs`, we now have a list of CropObject lists for each of the 140 documents.

Now that the dataset has been parsed, we can try to do some experiments! We can do for example symbol classification. Go check out the tutorial!

CHAPTER 4

Contents

CHAPTER 5

Indices and tables

- genindex
- modindex
- search