mung Documentation

Release 1.0.0

Jan Hajič jr. and Alexander Pacha

Nov 15, 2023

CONTENTS

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

https://github.com/OMR-Research/muscima-pp/

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 authors are Jan Hajič jr and Alexander Pacha. You can contact them at:

```
alexander.pacha@tuwien.ac.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/OMR-Research/mung

CHAPTER

ONE

REQUIREMENTS

Python 3.6, otherwise nothing beyond the `requirements.txt` file: `lxml` and `numpy`. If you want to apply pitch inference, you should also get `music21`.

CHAPTER

TWO

INSTALLATION

If you have pip, just run:

```
pip install mung
```

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

CHAPTER

THREE

FIRST STEPS

Let’s first download the dataset:

```
curl https://github.com/OMR-Research/muscima-pp/releases/download/v2.0/MUSCIMA-pp_v2.0.
↪zip > MUSCIMA-pp_v2.0.zip
unzip MUSCIMA-pp_v2.0.zip
cd MUSCIMA-pp_v2.0
```

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 mung.io import read_nodes_from_file
>>> node_filenames = [os.path.join('data', 'nodes_with_staff_annotations', f) for f in␣
↪os.listdir('data/nodes_with_staff_annotations')]
>>> docs = [read_nodes_from_file(f) for f in node_filenames]
>>> len(docs)
140
```

In `docs`, we now have a list of Node 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

FOUR

CONTENTS

CHAPTER FIVE

INDICES AND TABLES

- genindex
- modindex
- search