CiteXtract

Release 0.0.3

Jul 15, 2019

Contents:

1 Getting started **3**
- 1.1 Installation . . . 3
- 1.2 Extracting references . . . 3
- 1.3 Extracting titles . . . 3
- 1.4 Converting arXiv PDF to text . . . 4
- 1.5 Further reading . . . 4

2 Changelog **11**
- 2.1 0.0.3 . . . 11
- 2.2 0.0.2 . . . 11
- 2.3 0.0.1 . . . 11

3 Indices and tables **13**

Python Module Index **15**

Index **17**

The goal of CiteXtract is to bring structure to the references found on ArXiv papers. In order to start with CiteXtract, continue to the *getting started* page.

CHAPTER 1

Getting started

1.1 Installation

The installation of CiteXtract is done by the following command:

```
pip install citextract
```

CiteXtract is automatically tested on Python 3.5 or newer.

1.2 Extracting references

In order to extract references, the following code is used:

```
from citextract.models.refxtract import RefXtractor

refxtractor = RefXtractor().load()
text = """This is a test sentence.\n[1] Jacobs, K. 2019. This is a test title. In␣
↪Proceedings of Some Journal."""
refs = refxtractor(text)
print(refs)
```

This might produce the following output:

```
['[1] Jacobs, K. 2019. This is a test title. In Proceedings of Some Journal.']
```

In the code, the RefXtract model is initialized and the model parameters are downloaded from the internet. Then, the model is executed on an example text. It returns a list of the found references in the text.

1.3 Extracting titles

In order to extract titles from references, the following code is used:

```
from citextract.models.titlextract import TitleXtractor

titlextractor = TitleXtractor().load()
ref = """[1] Jacobs, K. 2019. This is a test title. In Proceedings of Some Journal."""
title = titlextractor(ref)
print(title)
```

This might produce the following output:

```
'This is a test title.'
```

In the code, the TitleXtract model is initialized and the model parameters are downloaded from the internet. Then, the model is executed on an example text. It returns a string of the found title in the reference.

1.4 Converting arXiv PDF to text

In order to get content for the RefXtract model, one can download a PDF from arXiv by using the following code:

```
from citextract.utils.pdf import convert_pdf_url_to_text

pdf_url = 'https://arxiv.org/pdf/some_file.pdf'
text = convert_pdf_url_to_text(pdf_url)
```

1.5 Further reading

The module documentation contains pointers to the different classes and methods that can be used.

1.5.1 citextract package

Subpackages

citextract.models package

Submodules

citextract.models.refxtract module

RefXtract package.

class citextract.models.refxtract.**BiRNN**(*input_size*, *hidden_size*, *num_layers=1*, *num_classes=2*, *device=None*)

Bases: sphinx.ext.autodoc.importer._MockObject

Bidirectional RNN model.

forward(*x*)

Forward-propagate the given input.

Parameters **x** (*torch.Tensor*) – The tensor of size [batch_size, sequence_length, input_size] to forward-propagate.

Returns The output, which has a shape of [batch_size, sequence_length, num_classes].

Return type torch.Tensor

class citextract.models.refxtract.**RefXtractPreprocessor**(*device=None*)

Bases: object

Preprocessor class for preprocessing textual data.

get_vocab_size()

Compute the size of the vocabulary.

Returns Size of the vocabulary.

Return type int

map_char(*char*)

Map a given character to a normalized class representant.

Parameters **char** (*str*) – The char to map.

Returns The mapped character.

Return type str

mapped_char_to_id(*mapped_char*)

Map a character to an numerical identifier.

mapped_char [str] The mapped character that should be converted to its numerical representation.

Returns The numerical representation of the character.

Return type int

class citextract.models.refxtract.**RefXtractText**(*text*, *idx*)

Bases: object

Simple helper class which contains the text and char indices of a given input.

class citextract.models.refxtract.**RefXtractor**(*model=None*, *preprocessor=None*, *device=None*)

Bases: object

RefXtractor class.

load(*model_uri=None*, *ignore_cache=False*)

Load model parameters from the internet.

Parameters

- **model_uri** (*str*) – The model URI to load from.
- **ignore_cache** (*bool*) – When true, all caches are ignored and the model parameters are forcefully downloaded.

Returns The wrapper itself.

Return type *RefXtractor*

citextract.models.refxtract.**build_refxtract_model**(*preprocessor*, *embed_size=128*, *hidden_size=128*, *device=None*)

Build an instance of the RefXtract model.

Parameters

- **preprocessor** (RefXtractPreprocessor) – The preprocessor to use.
- **embed_size** (*int*) – The number of embedding neurons to use.

- **hidden_size** (*int*) – The number of hidden neurons to use.
- **device** (*torch.device*) – The device to compute on.

Returns A RefXtract model instance.

Return type torch.nn.modules.container.Sequential

citextract.models.refxtract.**extract_references**(*text*, *preprocessor*, *model*)

Extract references from a given text.

Parameters

- **text** (*str*) – The text to extract the references from.
- **preprocessor** (RefXtractPreprocessor) – The preprocessor to use.
- **model** (*torch.nn.modules.container.Sequential*) – The model to use.

Returns A list containing the found references.

Return type list

citextract.models.refxtract.**preprocess_reference_text**(*text*)

Preprocess a PDF text.

Parameters **text** (*str*) – The text (possibly from a converted PDF) to preprocess.

Returns

A tuple consisting of the following elements: - has_reference_section : A boolean which is true when the text contained the string 'reference'

(not case-sensitive), false otherwise.

- reference_section : A string containing the reference section.
- non_reference_section : A string containing the text which was not in the reference section.

Return type tuple

citextract.models.titlextract module

The TitleXtract model.

class citextract.models.titlextract.**TitleTagging**(*input_size*, *hidden_size*, *n_layers*, *n_classes*, *device*)

Bases: sphinx.ext.autodoc.importer._MockObject

TitleTagging model.

forward(*x*)

Forward-propagate the input data.

Parameters **x** (*torch.Tensor*) – The input tensor of size (batch_size, sequence_length, input_size).

Returns The output tensor of size (batch_size, sequence_length, n_classes).

Return type torch.Tensor

class citextract.models.titlextract.**TitleXtractPreprocessor**(*device=None*)

Bases: object

TitleXtract preprocessor.

map_text_chars(*text*)
Map text to numerical character representations.

Parameters **text** (*str*) – The text to map.

Returns The tensor representing the mapped characters.

Return type torch.Tensor

map_text_targets(*text*, *title*)
Align and map the targets of a text.

Parameters

- **text** (*str*) – The text to map.
- **title** (*str*) – The title (substring of the text) to map.

Returns A tensor representing the characters of the text for which an element is 1 if and only if a character is both represented by the text and by the title, 0 otherwise.

Return type torch.Tensor

class citextract.models.titlextract.**TitleXtractor**(*model=None*, *preprocessor=None*, *device=None*)

Bases: object

TitleXtractor wrapper class.

load(*model_uri=None*, *ignore_cache=False*)
Load model parameters from the internet.

Parameters

- **model_uri** (*str*) – The model URI to load from.
- **ignore_cache** (*bool*) – When true, all caches are ignored and the model parameters are forcefully downloaded.

Returns The wrapper itself.

Return type *TitleXtractor*

citextract.models.titlextract.**build_titlextract_model**(*preprocessor*, *embed_size=32*, *hidden_size=64*, *device=None*)

Build an instance of the TitleXtract model.

Parameters

- **preprocessor** (TitleXtractPreprocessor) – The preprocessor to use.
- **embed_size** (*int*) – The number of embedding neurons to use.
- **hidden_size** (*int*) – The number of hidden neurons to use.
- **device** (*torch.device*) – The device to compute on.

Returns A RefXtract model instance.

Return type torch.nn.modules.container.Sequential

Module contents

Model definitions for the CiteXtract project.

citextract.utils package

Submodules

citextract.utils.model module

Model utilities.

`citextract.utils.model.`**`load_model_params`**(*model*, *model_name*, *model_uri*, *ignore_cache=False*, *device=None*)

Load model parameters from disk or from the web.

Parameters

- **model** (*torch.nn.modules.container.Sequential*) – The model instance to load the parameters for.
- **model_name** (*str*) – The name of the model which should be loaded.
- **model_uri** (*str*) – Part of the URL or full URL to the model parameters. If not specified, then the latest version is pulled from the internet.
- **ignore_cache** (*bool*) – When true, all caches are ignored and the model parameters are forcefully downloaded.
- **device** (*torch.device*) – The device to use.

Returns The loaded PyTorch model instance.

Return type torch.nn.modules.container.Sequential

Raises `ValueError` – When the model name is not supported.

citextract.utils.pdf module

PDF utilities for converting PDF to a usable format.

`citextract.utils.pdf.`**`convert_pdf_file_to_text`**(*path*)

Convert a PDF file to text.

Parameters **path** (*str*) – Path to the PDF file.

Returns The text found in the PDF file.

Return type str

`citextract.utils.pdf.`**`convert_pdf_url_to_text`**(*pdf_url*)

Convert a PDF URL to text.

Parameters **pdf_url** (*str*) – The URL to parse.

Returns The text which was found in the PDF document.

Return type str

Module contents

Utilities for the CiteXtract project.

Module contents

CiteXtract - Bringing structure to the papers on ArXiv.

CHAPTER 2

Changelog

2.1 0.0.3

- Bugfix of filename clashes for PDF to text method.

2.2 0.0.2

- Implementation of the core features.
- Implementation of the PDF utilities.
- Added CircleCI support.
- Added Docker Cloud support.
- Added ReadTheDocs support.

2.3 0.0.1

- Initial version with no features.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

citextract, 9
citextract.models, 7
citextract.models.refxtract, 4
citextract.models.titlextract, 6
citextract.utils, 8
citextract.utils.model, 8
citextract.utils.pdf, 8

Index

B

BiRNN (*class in citextract.models.refxtract*), 4
build_refxtract_model() (*in module citextract.models.refxtract*), 5
build_titlextract_model() (*in module citextract.models.titlextract*), 7

C

citextract (*module*), 9
citextract.models (*module*), 7
citextract.models.refxtract (*module*), 4
citextract.models.titlextract (*module*), 6
citextract.utils (*module*), 8
citextract.utils.model (*module*), 8
citextract.utils.pdf (*module*), 8
convert_pdf_file_to_text() (*in module citextract.utils.pdf*), 8
convert_pdf_url_to_text() (*in module citextract.utils.pdf*), 8

E

extract_references() (*in module citextract.models.refxtract*), 6

F

forward() (*citextract.models.refxtract.BiRNN method*), 4
forward() (*citextract.models.titlextract.TitleTagging method*), 6

G

get_vocab_size() (*citextract.models.refxtract.RefXtractPreprocessor method*), 5

L

load() (*citextract.models.refxtract.RefXtractor method*), 5
load() (*citextract.models.titlextract.TitleXtractor method*), 7
load_model_params() (*in module citextract.utils.model*), 8

M

map_char() (*citextract.models.refxtract.RefXtractPreprocessor method*), 5
map_text_chars() (*citextract.models.titlextract.TitleXtractPreprocessor method*), 6
map_text_targets() (*citextract.models.titlextract.TitleXtractPreprocessor method*), 7
mapped_char_to_id() (*citextract.models.refxtract.RefXtractPreprocessor method*), 5

P

preprocess_reference_text() (*in module citextract.models.refxtract*), 6

R

RefXtractor (*class in citextract.models.refxtract*), 5
RefXtractPreprocessor (*class in citextract.models.refxtract*), 5
RefXtractText (*class in citextract.models.refxtract*), 5

T

TitleTagging (*class in citextract.models.titlextract*), 6
TitleXtractor (*class in citextract.models.titlextract*), 7
TitleXtractPreprocessor (*class in citextract.models.titlextract*), 6