trec-car-tools

Release 1.0

Mar 03, 2022

Contents

1 Getting started **3**

2 Reading the dataset **5**

3 Basic types **7**

4 The `Page` type **9**
4.1 Types of pages 11

5 Page structure **13**

6 Paragraph contents **17**

7 Indices and tables **19**

Index **21**

This is the documentation for `trec-car-tools`, a Python 3 library for reading and manipulating the TREC Complex Answer Retrieval (CAR) dataset.

CHAPTER 1

Getting started

This library requires Python 3.3 or greater. It can can be installed with `setup.py`

```
python3 ./setup.py install
```

If you are using Anaconda, install the `cbor` library for Python 3.6:

```
conda install -c laura-dietz cbor=1.0.0
```

Once you have installed the library, you can download a dataset and start playing.

CHAPTER 2

Reading the dataset

The TREC CAR dataset consists of a number of different exports. These include,

- Annotations files (also called "pages files") contain full Wikipedia pages and their contents
- Paragraphs files contain only paragraphs disembodied from their pages
- Outlines files contain only the section structure of pages and no textual content

To read an annotations file use the `iter_annotations()` function:

`trec_car.read_data.`**`iter_annotations`**(*file*)
Iterate over the *Page*s of an annotations file.

Return type typing.Iterator[*Page*]

For instance, to list the page IDs of pages in a pages file one might write

```
for page in read_data.iter_annotations(open('train.test200.cbor', 'rb')):
    print(page.pageId)
```

Likewise, to read a paragraphs file the `iter_paragraphs()` function is provided

`trec_car.read_data.`**`iter_paragraphs`**(*file*)
Iterate over the *Paragraph*s of an paragraphs file.

Return type typing.Iterator[*Paragraph*]

To list the text of all paragraphs in a paragarphs file one might write,

```
for para in read_data.iter_paragraphs(open('train.test200.cbor', 'rb')):
    print(para.getText())
```

CHAPTER 3

Basic types

class trec_car.read_data.**PageName**
PageName represents the natural language "name" of a page. Note that this means that it is not necessarily unique. If you need a unique handle for a page use PageId.

class trec_car.read_data.**PageId**
A PageId is the unique identifier for a Page.

CHAPTER 4

The `Page` type

class trec_car.read_data.**Page**(*page_name*, *page_id*, *skeleton*, *page_type*, *page_meta*)

The name and skeleton of a Wikipedia page.

page_name

Return type *PageName*

The name of the page.

skeleton

Return type typing.List[*PageSkeleton*]

The contents of the page

page_type

Return type *PageType*

Type about the page

page_meta

Return type *PageMetadata*

Metadata about the page

flat_headings_list()

return Returns a flat list of headings contained by the `Page`.

Return type typing.List[*Section*]

get_text()

Include all visible text below this elements. Includes Captions of images, but no headings and no infoboxes. See *get_text_with_headings* for a version that includes headings.

get_text_with_headings(*include_heading=False*)

Include all visible text below this elements. While the heading of this element is excluded, headings of subsections will be included. Captions of images are excluded.

nested_headings()
Each heading recursively represented by a pair of `(heading, list_of_child_sections)`.

Return type typing.List[typing.Tuple[*Section*, typing.List[*Section*]]]

to_string()
Render a string representation of the page.

Return type str

class trec_car.read_data.**PageMetadata**(*redirectNames*, *disambiguationNames*, *disambiguationIds*, *categoryNames*, *categoryIds*, *inlinkIds*, *inlinkAnchors*, *wikiDataQid*, *siteId*, *pageTags*)

Meta data for a page

redirectNames

Return type *PageName*

Names of pages which redirect to this page

disambiguationNames

Return type *PageName*

Names of disambiguation pages which link to this page

disambiguationId

Return type *PageId*

Page IDs of disambiguation pages which link to this page

categoryNames

Return type str

Page names of categories to which this page belongs

categoryIds

Return type str

Page IDs of categories to which this page belongs

inlinkIds

Return type str

Page IDs of pages containing inlinks

inlinkAnchors
inlinkAnchor frequencies

rtype str

(Anchor text, frequency) of pages containing inlinks

wikidataQid

Return type str

Language and time independent Wikidata IDs (e.g. Q12345)

siteId

Return type str

SiteId (e.g. enwiki). The combination of WikidataQid and SiteId identifies a page in a wikipedia across time stamps. Note that PageName and PageId can change over time.

pageTags

Return type str

Template tags of pages, e.g. “Good article” or “Vital article”

4.1 Types of pages

class trec_car.read_data.**PageType**

An abstract base class representing the various types of pages.

Subclasses include

- *ArticlePage*
- *CategoryPage*
- *DisambiguationPage*
- *RedirectPage*

The abstact base class.

class trec_car.read_data.**ArticlePage**

class trec_car.read_data.**CategoryPage**

class trec_car.read_data.**DisambiguationPage**

class trec_car.read_data.**RedirectPage**(*targetPage*)

targetPage

Return type *PageId*

The target of the redirect.

CHAPTER 5

Page structure

The high-level structure of a `Page` is captured by the subclasses of `PageSkeleton`.

class trec_car.read_data.**PageSkeleton**
An abstract superclass for the various types of page elements. Subclasses include:

- *Section*
- *Para*
- *Image*

get_text()
Includes visible text of this element and below. Headings are excluded. Image Captions are included. Infoboxes are ignored. (For a version with headers and no captions see *get_text_with_headings*

get_text_with_headings(*include_heading=False*)
Include all visible text below this elements. While the heading of this element is excluded, headings of subsections will be included. Captions of images are excluded.

class trec_car.read_data.**Para**(*paragraph*)
Bases: *trec_car.read_data.PageSkeleton*

A paragraph within a Wikipedia page.

paragraph

Return type *Paragraph*

The content of the Paragraph (which in turn contain a list of *ParaBody*s)

get_text()
Includes visible text of this element and below. Headings are excluded. Image Captions are included. Infoboxes are ignored. (For a version with headers and no captions see *get_text_with_headings*

get_text_with_headings(*include_heading=False*)
Include all visible text below this elements. While the heading of this element is excluded, headings of subsections will be included. Captions of images are excluded.

class trec_car.read_data.**Section**(*heading*, *headingId*, *children*)
Bases: *trec_car.read_data.PageSkeleton*

A section of a Wikipedia page.

heading

Return type str

The section heading.

headingId

Return type str

The unique identifier of a section heading.

children

Return type typing.List[*PageSkeleton*]

The *PageSkeleton* elements contained by the section.

get_text()
Includes visible text of this element and below. Headings are excluded. Image Captions are included. Infoboxes are ignored. (For a version with headers and no captions see *get_text_with_headings*

get_text_with_headings(*include_heading=False*)
Include all visible text below this elements. While the heading of this element is excluded, headings of subsections will be included. Captions of images are excluded.

class trec_car.read_data.**List**(*level*, *body*)
Bases: *trec_car.read_data.PageSkeleton*

An list element within a Wikipedia page.

level

Return type int

The list nesting level

body
A *Paragraph* containing the list element contents.

get_text()
Includes visible text of this element and below. Headings are excluded. Image Captions are included. Infoboxes are ignored. (For a version with headers and no captions see *get_text_with_headings*

get_text_with_headings(*include_heading=False*)
Include all visible text below this elements. While the heading of this element is excluded, headings of subsections will be included. Captions of images are excluded.

class trec_car.read_data.**Image**(*imageurl*, *caption*)
Bases: *trec_car.read_data.PageSkeleton*

An image within a Wikipedia page.

caption

Return type str

PageSkeleton representing the caption of the image

imageurl

Return type str

URL to the image; spaces need to be replaced with underscores, Wikimedia Commons namespace needs to be prefixed

get_text()
Includes visible text of this element and below. Headings are excluded. Image Captions are included. Infoboxes are ignored. (For a version with headers and no captions see *get_text_with_headings*

get_text_with_headings(*include_heading=False*)
Include all visible text below this elements. While the heading of this element is excluded, headings of subsections will be included. Captions of images are excluded.

CHAPTER 6

Paragraph contents

class trec_car.read_data.**Paragraph**(*para_id*, *bodies*)
A paragraph.

get_text()
Get all of the contained text.

Return type str

class trec_car.read_data.**ParaBody**
An abstract superclass representing a bit of *Paragraph* content.

get_text()
Get all of the text within a *ParaBody*.

Return type str

class trec_car.read_data.**ParaText**(*text*)
Bases: *trec_car.read_data.ParaBody*

A bit of plain text from a paragraph.

text

Return type str

The text

get_text()
Get all of the text within a *ParaBody*.

Return type str

class trec_car.read_data.**ParaLink**(*page*, *link_section*, *pageid*, *anchor_text*)
Bases: *trec_car.read_data.ParaBody*

A link within a paragraph.

page

Return type *PageName*

The page name of the link target

pageid

Return type *PageId*

The link target as trec-car identifer

link_section

Return type str

Section anchor of link target (i.e. the part after the # in the URL), or `None`.

anchor_text

Return type str

The anchor text of the link

get_text()
Get all of the text within a *ParaBody*.

Return type str

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Index

A

anchor_text (*trec_car.read_data.ParaLink attribute*), 18
ArticlePage (*class in trec_car.read_data*), 11

B

body (*trec_car.read_data.List attribute*), 14

C

caption (*trec_car.read_data.Image attribute*), 14
categoryIds (*trec_car.read_data.PageMetadata attribute*), 10
categoryNames (*trec_car.read_data.PageMetadata attribute*), 10
CategoryPage (*class in trec_car.read_data*), 11
children (*trec_car.read_data.Section attribute*), 14

D

disambiguationId (*trec_car.read_data.PageMetadata attribute*), 10
disambiguationNames (*trec_car.read_data.PageMetadata attribute*), 10
DisambiguationPage (*class in trec_car.read_data*), 11

F

flat_headings_list() (*trec_car.read_data.Page method*), 9

G

get_text() (*trec_car.read_data.Image method*), 15
get_text() (*trec_car.read_data.List method*), 14
get_text() (*trec_car.read_data.Page method*), 9
get_text() (*trec_car.read_data.PageSkeleton method*), 13
get_text() (*trec_car.read_data.Para method*), 13
get_text() (*trec_car.read_data.ParaBody method*), 17
get_text() (*trec_car.read_data.Paragraph method*), 17
get_text() (*trec_car.read_data.ParaLink method*), 18
get_text() (*trec_car.read_data.ParaText method*), 17
get_text() (*trec_car.read_data.Section method*), 14
get_text_with_headings() (*trec_car.read_data.Image method*), 15
get_text_with_headings() (*trec_car.read_data.List method*), 14
get_text_with_headings() (*trec_car.read_data.Page method*), 9
get_text_with_headings() (*trec_car.read_data.PageSkeleton method*), 13
get_text_with_headings() (*trec_car.read_data.Para method*), 13
get_text_with_headings() (*trec_car.read_data.Section method*), 14

H

heading (*trec_car.read_data.Section attribute*), 14
headingId (*trec_car.read_data.Section attribute*), 14

I

Image (*class in trec_car.read_data*), 14
imageurl (*trec_car.read_data.Image attribute*), 14
inlinkAnchors (*trec_car.read_data.PageMetadata attribute*), 10
inlinkIds (*trec_car.read_data.PageMetadata attribute*), 10
iter_annotations() (*in module trec_car.read_data*), 5
iter_paragraphs() (*in module trec_car.read_data*), 5

L

level (*trec_car.read_data.List attribute*), 14
link_section (*trec_car.read_data.ParaLink attribute*), 18

List (*class in trec_car.read_data*), 14

N

nested_headings() (*trec_car.read_data.Page method*), 9

P

Page (*class in trec_car.read_data*), 9
page (*trec_car.read_data.ParaLink attribute*), 17
page_meta (*trec_car.read_data.Page attribute*), 9
page_name (*trec_car.read_data.Page attribute*), 9
page_type (*trec_car.read_data.Page attribute*), 9
pageid (*trec_car.read_data.ParaLink attribute*), 18
PageMetadata (*class in trec_car.read_data*), 10
PageSkeleton (*class in trec_car.read_data*), 13
pageTags (*trec_car.read_data.PageMetadata attribute*), 11
PageType (*class in trec_car.read_data*), 11
Para (*class in trec_car.read_data*), 13
ParaBody (*class in trec_car.read_data*), 17
Paragraph (*class in trec_car.read_data*), 17
paragraph (*trec_car.read_data.Para attribute*), 13
ParaLink (*class in trec_car.read_data*), 17
ParaText (*class in trec_car.read_data*), 17

R

redirectNames (*trec_car.read_data.PageMetadata attribute*), 10
RedirectPage (*class in trec_car.read_data*), 11

S

Section (*class in trec_car.read_data*), 13
siteId (*trec_car.read_data.PageMetadata attribute*), 10
skeleton (*trec_car.read_data.Page attribute*), 9

T

targetPage (*trec_car.read_data.RedirectPage attribute*), 11
text (*trec_car.read_data.ParaText attribute*), 17
to_string() (*trec_car.read_data.Page method*), 10
trec_car.read_data.PageId (*built-in class*), 7
trec_car.read_data.PageName (*built-in class*), 7

W

wikidataQid (*trec_car.read_data.PageMetadata attribute*), 10