mit-news-tools Documentation

Release 0.0.1

Emily Fan, Arun Wongprommoon

Sep 15, 2020

Quickstart

MIT News Tools is a package containing several tools to help with news processing. It was developed in Summer 2020 with the Tegmark Group at MIT. Some of its models have been integrated into the Tegmark Group's other projects, such as Improve The News. We have now released it as a package for public use!

See Installation and Usage to get started!

CHAPTER 1

Quick Start

1.1 Installation

Install the package with pip:

```
$ pip install mit-news-tools
```

Please install the packages that mit-news-tools depends on as well:

```
$ pip install pandas
$ pip install datefinder
$ pip install date_guesser
$ pip install confusables
$ pip install selenium
```

1.2 Usage

Extracting news urls from the news homepage:

```
from mitnewstools import extract_urls, filter_article_urls

# first download the html of the article, for instance, with newspaper3k
from newspaper import Article
homepage_url = "https://www.nytimes.com/"
art = Article(homepage_url)
art.download()
art_html = art.html

# extracting news urls
url_list = extract_urls(art.html, homepage_url)  # extracting all urls from the␣
↪homepage
news_url_list = filter_article_urls(url_list, homepage_url)  # extracting only news␣
↪articles
```

(continues on next page)

(continued from previous page)

Note that news_url_list will only contain articles from the New York Times. (Similarly if the homepage_url is https://www.washingtonpost.com/, then news_url_list will only contain articles from the Washington Post.)

Finding dates from a news article:

```
from mitnewstools import get_dates

# first download the html of the article, for instance, with newspaper3k
from newspaper import Article
art_url = "https://www.nytimes.com/2020/08/11/us/politics/pompeo-state-inspector-
↪general-saudi-weapons-civilian-casualties.html"
art = Article(art_url)
art.download()
art_html = art.html

date_published, date_modified = get_dates(art_html, art_url)
```

Removing accents or other non-ASCII characters in the article text:

```
from mitnewstools import asciify

# first download the text of the article, for instance, with newspaper3k
from newspaper import Article
art_url = "https://www.nytimes.com/2020/08/11/us/politics/pompeo-state-inspector-
↪general-saudi-weapons-civilian-casualties.html"
art = Article(art_url)
art.download()
art.parse()  # note that this example has this additional line
art_text = art.text  # since extracting the article text requires this step

ascii_article = asciify(art_text)
```

CHAPTER 2

mitnewstools package

2.1 Module contents

mitnewstools.**asciify**(*text: str*, *return_failed_chars=False*)

Takes a string and returns an ASCII version of it. If there is no suitable ASCII version of the string, it will be replaced by a space.

If return_failed_chars is True, it returns a tuple. The first element is the asciified string. The second element is a list of characters that failed to be converted into ASCII and instead were converted to spaces. example: "asciified string", [":)", ":—)"]

Parameters

- **text** – A string that you want to make sure is ASCII.
- **return_failed_chars** – If true, will return a list of characters that have failed to convert to ASCII

Returns an ASCII version of the input string; if return_failed_chars is True, it also returns a list of characters that failed to be converted into ASCII and instead were converted to spaces

mitnewstools.**selenium_download**(*url*, *driver=None*, *return_html=True*)

mitnewstools.**extract_news_urls_selenium**(*driver*, *match_file=None*) → pandas.core.frame.DataFrame

mitnewstools.**extract_base_url**(*url: str*, *endswithslash=True*) → str

Return a url that cuts off the item after the ? If endswithslash is True, returns a url that ends with a slash

mitnewstools.**extract_domain**(*url: str*) → str

Extracts the domain of a site. For instance, "https://www.economist.com/news/2020/06/19/frequently-asked-questions" becomes "economist.com"

mitnewstools.**extract_urls**(*html: str*, *base_url: str*) → list

Given the html and the url of a news homepage, return a list of urls that the homepage links to.

mitnewstools.**get_match_formula**(*domain*, *file=None*)

mitnewstools.**is_news_article**(*url: str*, *domain: str*, *match_formula=None*, *blacklist=None*) → bool

Parameters

- **url** – url of what is possibly an article.
- **domain** – the domain name of the newssite that the url should belong to
- **match_formula** – (optional) a list of regular expressions such that the url matches at least one of them
- **blacklist** – (optional) A list of regular expressions that the url should not follow

Returns True if the url is a news article from the same domain on the website

mitnewstools.**filter_article_urls**(*urls: list*, *domain: str*, *match_file=None*) → list

Parameters

- **urls** – list of urls
- **domain** – domain the url should be in
- **match_file** – (optional) file that contains a list of regular expressions for news articles

Returns a list of urls that are news articles and come from the specified domain

mitnewstools.**datefind_html**(*article_html: str*, *url: str*, *map_file=None*) → str

Given the html and url of a news article, return the date published in isoformat or an empty string if date cannot be found

mitnewstools.**datefind_json**(*article_html: str*) → dict

Given the html of a news article, return a dictionary with keys that starts with date, if found, such as datePublished, dateModified, or dateCreated. The values of the dictionary should be in isoformat. If such keys are not found, it returns an empty dictionary.

mitnewstools.**get_dates**(*article_html: str*, *url: str*) → tuple

Given the html and the url of the url, return the publication date and the modification date in isoformat as a tuple.

format is (date_published_iso, date_modified_iso)

("2020-05-27T21:59:25+01:00", "2020-05-28T18:34:13+01:00")

If either of the publication date or the modification date cannot be found, they will be a empty string in the tuple.

For instance, here is the example if the modification date was not found

("2020-05-27T21:59:25+01:00", "")

How it works:

1) Looks for date in a website's json.
2) If date not found, look for date in url.
3) If date still not found, look for date in html.
4) Use media cloud's dateguesser.

CHAPTER 3

Authors

3.1 Team

- Emily Fan
- Arun Wongprommoon
- Max Tegmark (Principal Investigator)

3.2 Contributors

- Jamie Fu (for read the docs support)

3.3 Contact Us

You can contact us with any questions at emilyfan [at] mit [dot] edu.

Python Module Index

m

Index