Accessing open research literature with Python

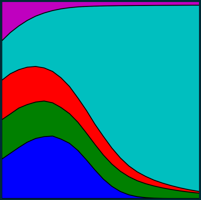
Nikoleta Glynatsi

2017-02

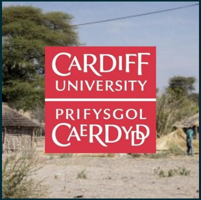
About me

CARDIFF UNIVERSITY

PRIFYSGOL CAERDYDD



Software
Sustainability
Institute



Definition

A **literature review** is a search and evaluation of the available literature in your given subject or chosen topic area.
-Royal Literary Fund

Definition

Web Scraping is a technique employed to extract large amounts of data from websites. - WebHarvy

- Literature Review;
- World Wide Web;
- Web Scraping;

Arcas

https://github.com/Nikoleta-v3/Arcas

Arcas

```
pip install arcas
```

Arcas

```
pip install arcas
```

```
git clone git@github.com:Nikoleta-v3/Arcas.git
python setup.py develop
```

Arcas

```
arcas_scrape -p arxiv -t "Prisoner’s Dilemma" -y 2014 -r 1
```

Arcas

```
arcas_scrape -p arxiv -t "Prisoner's Dilemma" -y 2014 -r 1
```

```
arcas_scrape -p ieee -t "Prisoner Dilemma" -a "Nowak" -y 2014
... -r 1 -s 2
```

Arcas

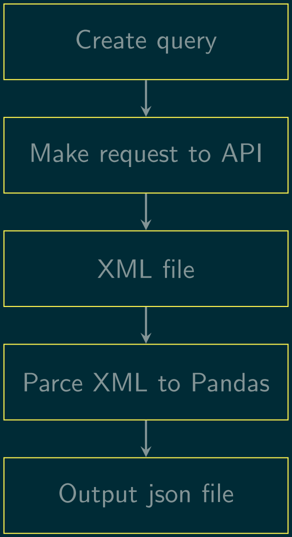
```
arcas_scrape -p arxiv -t "Prisoner’s Dilemma" -y 2014 -r 1
```

```
arcas_scrape -p ieee -t "Prisoner Dilemma" -a "Nowak" -y 2014
... -r 1 -s 2
```

```
arcas_scrape -p ieee -b "game theory" -t "Prisoner Dilemma"
... -a "Nowak" -y 2014 -r 1 -s 2
```

Results

```
{
"key":{"0":"Deng2014","1":"Deng2014","2":"Deng2014"},
"unique_key":{"0":"3369e749ce1c92062806e7fa3c41c90e",
              "1":"3369e749ce1c92062806e7fa3c41c90e",
              "2":"3369e749ce1c92062806e7fa3c41c90e"},
"title":{"0":"Generalized prisoner’s dilemma",
         "1":"Generalized prisoner’s dilemma",
         "2":"Generalized prisoner’s dilemma"},
"author":{"0":"Xinyang Deng","1":"Qi Liu","2":"Yong Deng"},
"abstract":{"0":"  Prisoner’s dilemma has been ...",
            "1":"  Prisoner’s dilemma has been ...",
            "2":"  Prisoner’s dilemma has been ..."},
"date":{"0":2014,"1":2014, "2":2014},
"journal":{"0":"arXiv","1":"arXiv","2":"arXiv"},
"provenance":{"0":"arXiv","1":"arXiv","2":"arXiv"}
}
```



Create query

Make request to API

XML file

Parce XML to Pandas

Output json file

```
from arcas import *

def get_arguments(api, word, count):
    arguments = [{’-a’: None, ’-b’: word, ’-s’: None,
                  ’-r’: count, ’-y’: None, ’-t’: None},
    return arguments

def main_program(arguments):
    parameters = pp.parameters_fix(arguments=arguments)
    url = pp.create_url_search(parameters=parameters)
    response = pp.make_request(url)
    root = pp.get_root(response)
    article = pp.parse(root)
    return article
```

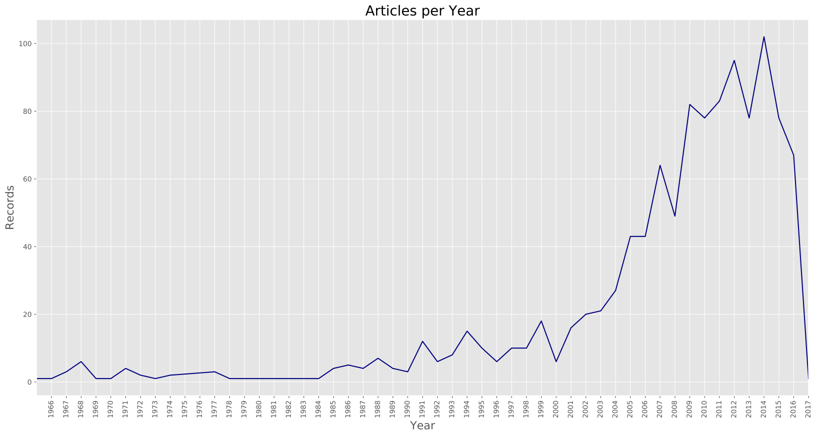
```
from arcas import *

def get_arguments(api, word, count):
    arguments = [{’-a’: None, ’-b’: word, ’-s’: None,
                  ’-r’: count, ’-y’: None, ’-t’: None},
    return arguments

def main_program(arguments):
    parameters = pp.parameters_fix(arguments=arguments)
    url = pp.create_url_search(parameters=parameters)
    response = pp.make_request(url)
    root = pp.get_root(response)
    article = pp.parse(root)
    return article
```

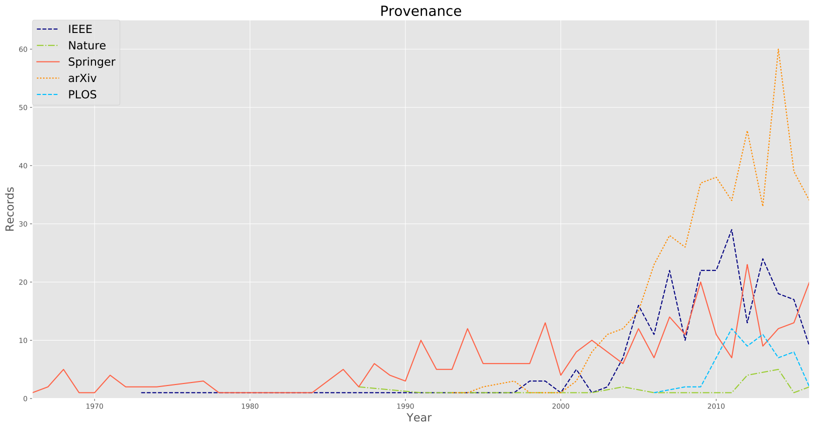
```
words = ["prisoner’s dilemma", "prisoners evolution", "prisoner dilemma", ...]
apis = {"ieee": Ieee, "nature": Nature, "arxiv": Arxiv, "springer": Springer, "plos": Plos}
list_apis = [’plos’, ’arxiv’, ’ieee’, ’nature’, ’springer’]
count = 10
```

```python
from arcas import *

def get_arguments(api, word, count):
    arguments = [{’-a’: None, ’-b’: word, ’-s’: None,
                  ’-r’: count, ’-y’: None, ’-t’: None},
    return arguments

def main_program(arguments):
    parameters = pp.parameters_fix(arguments=arguments)
    url = pp.create_url_search(parameters=parameters)
    response = pp.make_request(url)
    root = pp.get_root(response)
    article = pp.parse(root)
    return article
```

```python
words = ["prisoner’s dilemma", "prisoners evolution", "prisoner dilemma", ...]
apis = {"ieee": Ieee, "nature": Nature, "arxiv": Arxiv, "springer": Springer, "plos": Plos}
list_apis = [’plos’, ’arxiv’, ’ieee’, ’nature’, ’springer’]
count = 10
```

```python
for wr in words:
    for p in list_apis:
        pp = apis[p]()

        arguments = get_arguments(p, wr, count)
        raw_articles = main_program(arg)

        df = pp.to_dataframe(raw_article)
        pp.export(dfs, filename=’articles/{}-{}.json’.format(p, wr))
```


Articles per Year
Records
100
80
60
40
20
0
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
Year


Provenance
IEEE
Nature
Springer
arXiv
PLOS
Records
Year
0
10
20
30
40
50
60
1970
1980
1990
2000
2010

@NikoletaGlyn
https://github.com/Nikoleta-v3
ttps://github.com/Nikoleta-v3/Arcas