

1 Altmetrics meet my publications

Altmetrics is an alternative to simple citation counts of articles. Altmetrics looks at how your papers are mentioned in Tweets, google+, blog posts, news, how many Mendeley users have the article, etc... They are partnering with publishers to provide additional metrics on your papers.

You can put some Altmetric badges on your papers so you can see how they are doing. In this post, we scrape out my papers from my orcid page, and add Altmetric badges to them. This is basically just a little snippet of html code that will put the Altmetric donut in the citation, which has some information about the number of times each paper is tweeted, etc...

So, here is a python script that will print some html results. We print each title with the Altmetric donut.

```
1 import requests
2 import json
3
4 resp = requests.get("http://pub.orcid.org/0000-0003-2625-9232/orcid-works",
5                     headers={'Accept': 'application/orcid+json'})
6 results = resp.json()
7
8 data = []
9 TITLES, DOIs = [], []
10
11 badge = "<div data-badge-type='medium-donut' class='altmetric-embed' data-badge-details='right' data-doi='{doi}'></div>"
12 scopus_cite = "<img src='http://api.elsevier.com/content/abstract/citation-count?doi={doi}&httpAccept=image/jpeg' alt='Altmetric badge' data-doi='{doi}'>"
13 html = '<a href="http://dx.doi.org/{doi}">{title}</a>'
14
15 print '<ol>'
16 for i, result in enumerate( results['orcid-profile']['orcid-activities']
17                             ['orcid-works']['orcid-work']):
18     title = str(result['work-title']['title']['value'].encode('utf-8'))
19     doi = 'None'
20
21     for x in result.get('work-external-identifiers', []):
22         for eid in result['work-external-identifiers']['work-external-identifier']:
23             if eid['work-external-identifier-type'] == 'DOI':
24                 doi = str(eid['work-external-identifier-id']['value'].encode('utf-8'))
25
26     # AIP journals tend to have a \n in the DOI, and the doi is the second line. we get
27     # that here.
28     if len(doi.split('\n')) == 2:
29         doi = doi.split('\n')[1]
30
31     pub_date = result.get('publication-date', None)
32     if pub_date:
33         year = pub_date.get('year', None).get('value').encode('utf-8')
34     else:
35         year = 'Unknown'
36
37     # Try to minimize duplicate entries that are found
```

```

38     dup = False
39     if title.lower() in TITLES:
40         dup = True
41     if (doi != 'None'
42         and doi.lower() in DOIs):
43         dup = True
44
45     if not dup:
46         # truncate title to first 50 characters
47         print '<li>'
48         print(html.format(doi=doi, title=title) + badge.format(doi=doi) + scopus_cite.format(doi=doi) + '<br>\n')
49         print('</li>')
50
51     TITLES.append(title.lower())
52     DOIs.append(doi.lower())
53
54 print '</ol>'

```
