paper_selection

June 11, 2017

0.1 Sampling of papers from conferences

This Jupyter notebook shows the procedure used to select a sub-sample of the accepted papers from each conference.

0.1.1 Accepted conference papers

Sampling of papers is based on the listing of accepted papers at the following locations:
AAAI-14 http://www.aaai.org/Library/AAAI/aaai14contents.php
AAAI-16 http://www.aaai.org/Library/AAAI/aaai16contents.php
IJCAI-13 http://ijcai-13.org/program/accepted_papers
IJCAI-16 http://ijcai-16.org/index.php/welcome/view/accepted_papers

These listings were used to generate the files available in the ../data/ folder. Each conference is represented by a textfile containing the papers accepted to the conference's main and special tracks. Each line in the textfiles represent a paper, including its title and the authors. Example:

```
Causality based Propagation History Ranking in Social Networks  Zheng Wang, Chaokun Wang, Jishe
Intervention Strategies for Increasing Engagement in Volunteer-Based Crowdsourcing  Avi Segal,
```

Papers are available through AAAI Publications for all but IJCAI-16 (at the time of writing):
AAAI-14 http://www.aaai.org/ocs/index.php/AAAI/AAAI14/schedConf/presentations
AAAI-16 http://www.aaai.org/ocs/index.php/AAAI/AAAI16/schedConf/presentations
IJCAI-13 http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/schedConf/presentations
For IJCAI-16, see the proceedings at: http://www.ijcai.org/Proceedings/2016
First, the accepted papers are loaded from files.

```
In [1]: from glob import glob

        accepted_papers = {}
        track_files = glob('../data/accepted*'.format(dir))
        for file in track_files:
            conference = file.split('_')[-1]
            accepted_papers[conference] = []
            with open(file, 'r') as f:
                for line in f:
                    accepted_papers[conference].append(line)
```

The resulting dictionary accepted_papers contains a list of the accepted papers for each conference.

```
In [2]: for conference, papers in sorted(accepted_papers.items()):
            print('{conference} includes {papers} accepted papers.'.format(
                conference=conference, papers=len(papers)))
```

```
aaai-14 includes 398 accepted papers.
aaai-16 includes 548 accepted papers.
ijcai-13 includes 413 accepted papers.
ijcai-16 includes 551 accepted papers.
```

0.1.2 Selection

A sample population of 100 papers is selected from each conference using Python's pseudo-random number module. As per the documentation on random.sample "*The resulting list is in selection order so that all sub-slices will also be valid random samples.*" The seed is set to the unix timestamp for Jan 10 14:46:40 2017 UTC: 1484059600.

```
In [3]: import random
        random.seed(1484059600)

        k = 100
        samples = {}

        # The order is set explicitly due to originally not sorting
        # accepted_papers.items().
        conferences = ['aaai-16', 'aaai-14', 'ijcai-13', 'ijcai-16']

        for conference in conferences:
            samples[conference] = random.sample(accepted_papers[conference], k)
```

Note that when originally generating the samples, the dictionary was iterated by the use of Python 3's dict.items() view. The order is not guaranteed, and I forgot to sort the iteration so repeated runs of the code would generate the same populations. Therefore, the order has to be set explicitly as above to generate the original populations.

The generated random samples are permanently stored to files in the ../data/ directory (Github: https://github.com/sidgek/msoppgave/tree/master/data/.

```
In [4]: for conference, papers in samples.items():
            outputfile = '../data/sampled_{conference}'.format(conference=conference)
            with open(outputfile, 'w') as f:
                for line in papers:
                    f.write(line)
```