Knowledge Acquisition for Next Generation Statement Map

Author: Eric Nichols, eric@ecei.tohoku.ac.jp

Tools

- instances2matrix.py: creates a matrix of co-occurence counts between relation pattern x arguments in mongodb from input instances

Instances

Format

Instances have the following tab-delimited format:

- score: score representing weight * co-occurence count for instance
- loc: giving source and location of instance
- rel: containing relation pattern
- argc: giving argument count
- argv: tab-delimited list of arguments as strings

Example

```
1.0\treverb_clueweb_tuples-1.1.txt:30:10-11\tARG1
acquired ARG2\t2\Google\tYouTube
```

Co-occurence Matrix

Format

The co-occurence matrix collection has the following fields:

- rel: relation pattern

- arg1: first argument
- …
- argn: nth argument
- score: score for rel x args tuple

Naming Scheme

Instances of differing argument count are stored in separate mongodb collections with names formatted as `<collection>_<argc>`. E.g. if a collection `clueweb` has instances with argument counts of 1, 2, and 3, then the following collection would be created:

- `clueweb_1`
- `clueweb_2`
- `clueweb_3`

Indexing

It is indexed for fast look up of rel, args, and (rel,args) tuples.

TO-DO

- should strings be binarized?
- cache co-occurance counts to separate databases
- finish map-reduce implementation of PMI and cache to separate database