**Agreement stuff rescued from email.**

**Note that Catherine’s work is also in drop box. Moving to github.**

Two things: The first is identifying software mentions. I think I calculated that, based on percent agreement code in Java. The reasoning is taken from one of Kevin’s content analysis papers.

The second was allocating codes to mentions. That’s what Catherine discusses below.

Hi James,

I had difficulty using the Cohen's kappa found in the IRR package because my data did not appear to be set up properly, so I used the Cohen's kappa found in the Concord package of R instead.

Here's the description of it:<http://rss.acs.unt.edu/Rdoc/library/concord/html/cohen.kappa.html>

Below are the agreement stats for each of the codes. One thing about the Concord Cohen's kappa dealie is that it threw an error for the codes "in-text\_mention," "software\_used," and any other code that was all 1s (in other words, where we both agreed completely and we both included the code for every single one). I think the error is probably because the data didn't show any evidence of a second category, so R didn't know how to handle it -- otherwise it would appear that the Cohen's kappa would be 1 for those codes. I'm sure it's just a matter of adding some sort of parameter to indicate that there is another category, it's just not chosen at all, but since it's getting late I figure I'd send this now and explore a little bit on my own.

Under /Users/Catherine/Dropbox/Shared\ Software\ Citation\ Folder/ttl-coding/CatherineCoding/code\_dataframes you will be able to find the csv files I read into R, a word doc of the text below called ReliabilityStatsParagraph, and a file called Cohen\_Kappa\_R\_code.txt that has the pertinent snippets of my R code. Thanks!

-- Catherine

-----------------

We measured interrater reliability using a subset of 12 articles selected at random from our general pool of 90 articles. The two authors, who also performed the coding, each coded the 12 articles in the subsample, and reliability statistics were calculated based on agreement shown on that subsample. Disagreements about coding decisions were resolved by discussing any discrepancies until both coders came to agreement. Once agreement had been reached and disagreements resolved, one coder conducted the coding of the rest of the general sample. In total, it took approximately \_\_\_ training hours to reach the reliability levels reported below.

Cohen’s Kappa was calculated on all codes using the Concord package of the R statistical program. We chose Cohen’s Kappa as our agreement measure because \_\_\_\_\_\_\_\_\_

The intercoder reliability level for each code is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Cohen’s Kappa | Siegel & Castellan | Byrt, Bishop and Carlin (2\*PA-1) |
| configuration\_details | k = 0.873892  Z = 52.4982  p = 0 | k = 0.264368  Z = 0.676626  p = 0.249322 | k = 0.75 |
| creator | k = 1  Z = 19.3014  p = 0 | k = 1  Z = 4.66026  p = 1.57904e-06 | k = 1 |
| date | k = 1  Z = 19.3014  p = 0 | k = 1  Z = 2.55941  p = 0.00524247 | k = 1 |
| has\_reference | k = 0.930159  Z = 13.6883  p = 0 | k = 0.869048  Z = 4.63381  p = 1.79501e-06 | k = 0.878788 |
| in-text\_mention | Error (no 0s) |  |  |
| software\_name | Error (no 0s) |  |  |
| software\_not\_used | Error (no 0s) |  |  |
| software\_used | k = -0.015873  Z = -0.0229902  p = 0.509171 | k = -0.0322581  Z = -0.0463257  p = 0.518475 | k = 0.875 |
| url | k = 1  Z = 180.931  p = 0 | k = 1  Z = 1.4361  p = 0.0754876 | k = 1 |
| version\_number | k = 1  Z = 17.1953  p = 0 | k = 1  Z = 4.91015  p = 4.5503e-07 | k = 1 |

For further information on the coding process, interested parties can contact either of the authors.

Note: the low value for software\_used is due to prevalence (essentially there is completely disagreement on that code because the negative only occurs twice, and both coders don’t ever apply the negative). Thus chose to use the Byrt correction for that.

**LIST OF CODES:**

in-text\_mention

"Mention of software in the text part of the paper"

reference

"A software relevant reference"

* If multiple references are present in the coding unit and refer to the same piece of software, multiple has\_reference codes should be attached to the same in-text\_mention, not broken up.

software\_used

"For mentions of software that was used in the research"

software\_not\_used

"For mentions of software that the authors didn't use (e.g., they discuss why they didn't use it)"

* This code should be used for software that the authors considered (e.g., the competitors of the software they used).

software\_name

rdfs:label "whatever the name is"

version\_number

rdfs:label "whatever the version number is. If use date as version number include both codes"

url

rdfs:label "whatever the url is"

date

"A date used to indicate a version of the software. Not the date of the paper or reference. If use date as version number include both codes"

rdfs:label "whatever the date is"

creator

"A mention of the creator of the software, don't include names as citation anchors. Haven't decided what to do about references here yet. Example of where to use: Software, written by xyz, was used for ... 'coupled to MetaMorph imaging software (Universal Imaging Corporation, Downingtown, PA).' Code just the company name"

rdfs:label "whoever the creator is"

memo

"Any interesting issues that come up in the paper"

configuration\_details

"Any mention of configuration of the software."

* James coded formulas as a configuration detail.

For bioj-cited:xxx coding:

software\_publication

“Reference to a paper primarily detailing the software itself”

domain\_publication

“Reference to a biology research paper (cited because it uses or introduces the software”

users\_guide

“Reference to a user’s manual for the software”

project\_page

“URL for the software’s online presence - downloads, orders, FAQs”

project\_name

“Descriptive data about the software”

citec:creator - unlike in the rest of the text, code the author names of the paper, and include the full string of multiple authors as a single value for a :creator code.

Example for coding references:

bioj-cited:1993-McCune rdf:type bioj:selection ;

bioj:full\_quote """MCCUNE, B. 1993. Multivariate analysis on the PC-ORD system. Oregon State University, Corvallis, Oregon. USA. http:// bioag.byu.edu /zoology/crandall\_lab"""" # 1993-McCune tag on paper in Mendeley ;

ca:isTargetOf [ rdf:type ca:CodeApplication ;

ca:hasCoder "jhowison" ;

ca:appliesCode citec:software\_publication ; #domain\_publication, #users\_guide, #project\_page, #project\_name. Go and look at the type of thing that is at the 'end' of the reference.

] ;

ca:isTargetOf [ rdf:type ca:CodeApplication ;

ca:hasCoder "jhowison" ;

ca:appliesCode [ rdf:type citec:creater ;

rdfs:label "McCune, B" ]

] ;

ca:isTargetOf [ rdf:type ca:CodeApplication ;

ca:appliesCode [ rdf:type citec:url ;

rdfs:label "http:// bioag.byu.edu /zoology/crandall\_lab" ] ;

] ;

For journals in the set:

bioj:JOURNAL\_ABBR rdf:type bioj:journal ;

dc:urn dc:ISSN ;

dc:title "Title Text" ;

bioj:hasAuthorInstructions "url" ;

bioj:hasSoftwarePolicy "text" ;

bioj:hasSoftwarePolicy "url" ;

.

Where “ISSN” is in the form ####-#### and the two urls are the same unless the policy for citing software is in a different document than the author guidelines landing page.

-------------

**NOTES ON CODING**

In situations of multiple pieces of software mentioned in one sentence, that is coded as three (or however many) separate selections, each with their associated codes (meaning three separate mentions). The selection is the thing that gets the codes applied to, which is why we need three different anonymous [ ] blocs rather than applying three mention codes to the same one. It's fine for the selections to have identical full\_quotes.

Try to match selections to semantically different uses of the software\_name. In practice this seems to mean favoring larger selections, e.g., paragraphs, since that's the way that the authors seem to write about the software.

Catherine says: In a few instances models and software programs bearing the same name are discussed together and perhaps interchangeably (paste “geNorm” example here from 2010-05-BMC\_MOL\_BIOL). In this case I erred on the side of coding it as a software\_mention, except when it was clear to me that they were definitely not referring to the software (specifically, when a different piece of software -- qBasePlus -- was said to \*use\* the “geNorm model,” which precluded any interpretation of geNorm as a piece of software in this case).

---------------

Probably should code for the absence of a reference. The Open World Assumption in RDF means that leaving out codes doesn’t necessarily mean that they don’t apply.

Should a sentence with three references be coded as three separate in-text\_mentions? Probably, since we’d do that with three software\_names. (**Answer: No. A mention of a piece of software with three references should be coded as an in-text mention with three has\_reference codes attached to it.)**

---------------

Does the following example count as containing configuration details?:

**"IC50s were measured by fitting data to a three-parameter logistic curve using the formula Y=a+(b-a)/(1+10^(X-c)) (BioDataFit, Chang Bioscience)."** Is specifying the formula used in the software the same as specifying the settings? (James coded it as configuration\_details)

---------------

**CITATION FUNCTIONS**

Is it identifiable (from the info provided, can we confirm the specific software being referenced?

ca:isTargetOf

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:identifiable ;

] ;

ca:isTargetOf

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:unidentifiable ;

] ;

Is it findable? Can we find a project page or other official information regarding the program?

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:findable ;

] ;

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:unfindable ;

] ;

Can we find the specific version listed in the article? (if no version listed, then no)

ca:isTargetOf

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:findable\_version ;

] ;

ca:isTargetOf

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:unfindable\_version ;

] ;

Is it possible to access the program now? If so, is it free or for purchase? (I’ve been coding those with trial/evaluation versions and pay full versions as for-purchase)

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:access\_free ;

] ;

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:access\_purchase ;

] ;

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:no\_access ;

] ;

Is the source code available, either with the binary program or as a separate purchase?

ca:isTargetOf

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:source\_available ;

] ;

ca:isTargetOf

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:source\_unavailable ;

] ;

Do the creators provide permission to modify the program? (If no permission listed, or based on prior request to author, then no)

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:permission\_modify ;

] ;

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:prohibited\_modify ;

] ;

If the project page includes suggested citations, does the article use one of them?

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:matchesPreferredCite ;

] ;

ca:isTargetof

[ rdf:type ca:CodeApplication ;

ca:hasCoder "jbullard" ;

ca:appliesCode citec:missesPreferredCite ;

] ;

ca:memo "Explanation of missing the preferred citation - is the preferred citation older than the article?" ;