What's in our Backlogs?

A journey through Samvera Community Institutions' open issues with Natural Language Processing

presented by Anna Headley

project by Anna Headley and Eliot Jordan, Princeton University Library
What's next for you?
Question
What kinds of open issues do we have?

- Can we extract an interesting set of widely-desired features or widely-held use cases?
- Can we identify connections that might lead to collaboration across institutions?
Data Set
Collate a list of repositories

Dependency graph

Repositories that depend on hyrax

- CottageLabs / willow
  - Star: 5, Fork: 2

- RepoCamp / ucla2019-TeamB
  - Star: 0, Fork: 0

- RepoCamp / ucla2019-TeamC
  - Star: 0, Fork: 0

- RepoCamp / ucla2019-TeamA
  - Star: 0, Fork: 0
Let's use github's graphql api

**GraphQL API v4**

**DependencyGraphDependency**

This part of the schema is currently available for developers to preview. During this preview period, the API may change without any advance notice. Please see the Access to a Repositories Dependency Graph preview for more details.

**Note:** The GraphQL resources under preview cannot be accessed via the Explorer at this time.

A dependency manifest entry

i. **Fields**

**Fields**

`hasDependencies (Boolean!)`  
Does the dependency itself have dependencies?

`packageManager (String)`  
The dependency package manager
Collate a list of repositories

Dependency graph

Repositories that depend on hyrax

- CottageLabs / willow
  - Stars: 5
  - Forks: 2

- RepoCamp / ucla2019-TeamB
  - Stars: 0
  - Forks: 0

- RepoCamp / ucla2019-TeamC
  - Stars: 0
  - Forks: 0

- RepoCamp / ucla2019-TeamA
  - Stars: 0
  - Forks: 0
Acceptance criteria

- Code is hosted on github
- Github repository contains 1 or more issues
- Code powers a staff or public application at a specific institution or consortium
- Not in a samvera github organization
- Has a commit in the past 6 months
36 Repositories included in the data set

cul/ldpd-hyacinth  DigitalWPI/digitalwpi
curationexperts/laevigata  duke-libraries/ddr-public
curationexperts/tenejo  emory-libraries/dlp-curate
dibbs-vdc/ccql  galterlibrary/digital-repository
digital-york/arch1  gwu-libraries/scholarspace-hyrax
digital-york/ncelp  LafayetteCollegeLibraries/spot
digital-york/oasis  mlibrary/heliotrope
Collect and save the documents

```
def download_batch(cursor: , type:)
    response = @client.query
    query {
        query {
            repository(name: "#{repository}", owner: "#{organization}") {
                type{pagination_parameters(cursor: cursor)} {
                    edges {
                        cursor
                        node {
                            type_fields.downcase.to_sym
                        }
                    }
                    total_count
                }
            }
        }
        graphql
        response
    end
```
def issues_fields
  <<-FIELDS
    title
    bodyText
    comments (first: 100) {
      nodes {
        body
      }
    }
    number
    closed
    createdAt
  FIELDS
end
NLP Methods
The Google NLP APIs can

- Identify parts of speech
- Parse dates and contact information
- Identify corporate logos
- Perform sentiment analysis
- Categorize docs to a pre-defined list
- Train custom models to do document categorization based on a training set you provide.
Consult an expert

Thank you Rebecca Koeser, lead developer, and other helpful staff at Princeton's Center for Digital Humanities!
K-means clustering

**Clean data**

Stripped out all github usernames and created a stopwords list with institution-specific keywords that showed up in our clusters.

**_tokenize, stem, TF-IDF**

Used an nltk algorithm called WordNetLemmatizer to tokenize and stem the documents. Passed this tokenizer and our stopwords list with our documents into the SciKit TfidfVectorizer to get word frequency vectors.

**Cluster**

Pass the vectors to SciKit's k-means algorithm and piece the cluster numbers back together with the filenames so we can see what it did.
K-means clustering
Results
Cluster 0:
error test email job work run log message user server
229 issues in 28 repositories

Cluster 1:
page view link publisher line backtrace user add object admin
162 issues in 25 repositories

Cluster 2:
user work image need add use ingest item like resource
1166 issues in 34 repositories

Cluster 3:
collection work page user add item object need metadata search
182 issues in 22 repositories

Cluster 4:
file work upload set thumbnail version user preservation need csv
214 issues in 25 repositories

Cluster 5:
search result advanced page term text user field like item
94 issues in 23 repositories

Cluster 6:
field metadata work value form display record data need collection
255 issues in 28 repositories

Cluster 7:
date embargo range year field collection work facet visibility need
75 issues in 20 repositories
Helpful clusters
Cluster 39: Fixity checks
  task rake fixity check file run cron running fedora job
  29 issues in 11 repositories

Cluster 9: Full text search
  search text result searching pdf document extracted term full fulltext
  26 issues in 14 repositories
Cluster 12: User roles
  user dashboard press role page registered admin menu hyrax login
  30 issues in 10 repositories

Cluster 15: bagit
  bag visibility file validation work extracted import export archival badge
  28 issues in 10 repositories
Cluster 16: Thumbnail images, representative images
  thumbnail file image set blank representative fileset work resource manager
  35 issues in 13 repositories

Cluster 17: Embargoes
  embargo visibility expired work expiring notification embargoed object prod rake
  26 issues in 11 repositories
Cluster 58: IIIF
image viewer riiif iiif 308 tiff jp2 f derivative work
49 issues in 18 repositories

Cluster 30: more IIIF
manifest url iiif link viewer collection mirador
sammelband image like
36 issues in 13 repositories
Cluster 26: Blacklight range limit
blacklight limit year fix range search autocomplete
view facet byte
16 issues in 9 repositories

Cluster 36: date facets
facet date year sort result search decade az show field
25 issues in 14 repositories
Cluster 23: Linked data, SPARQL
   allow sufia rdf format triple regular user thing caption sparql
   23 issues in 8 repositories

Cluster 27: User account interactions
   email password contact address reset send user notification form department
   36 issues in 14 repositories
Cluster 49: Controlled vocabularies

term vocabulary controlled field search local use query deprecated json

28 issues in 13 repositories

Cluster 43: Controlled vocabularies for places

uris controlled geonames string osu vocabulary value move place location

16 issues in 5 repositories
Cluster 63: File characterization
  fit config characterization file use update performance ffmpeg script currently
  22 issues in 14 repositories

Cluster 60: Front end
  label location uri accessibility element input add form field content
  44 issues in 16 repositories
Cluster 18: Workers and resque
  job worker run derivative fixity server resque error
  new queue
  26 issues in 12 repositories

Cluster 7: Deployment concerns
  server monitoring cap production deploy deployment
task capistrano staging add
  29 issues in 13 repositories
Cluster 34: Universal Viewer
  object viewer video audio universal user like view digitized able
  44 issues in 11 repositories

Cluster 44: Browse Everything
  google drive file browseeverything meta oauth dropbox content browse transcription
  19 issues in 9 repositories
Unhelpful clusters
Cluster 1: Institution specific language

form update get put base rail changelog 8 unpaywall beavernetes

15 issues in 10 repositories
Cluster 2: Too broad

error file 500 message log 404 fatal info work import

66 issues in 21 repositories
Cluster 8: Probably need more stop words
add use data button link title need work set like
307 issues in 33 repositories

Cluster 35: Very general within our domain
field metadata value collection form display data related dictionary add
86 issues in 21 repositories
Next steps
Automate data cleaning

- Find more stop words. Exclude any word that's only found in issues from a single repository.
- Automate removal of user names by checking the github api when we strip tokens beginning with `@`. 
Look more closely at clusters

For the clusters we identified, look at the actual issues that belong to them and see how cohesive they feel
Run and analyze issues and PRs together

- If we introduce more robust cleaning mechanisms, we could try clustering issues and PRs together to see whether we could match working code to backlog issues across institutions.
Is this helpful?
Final thoughts

We weren't able to get issues for every relevant project in the community.
Final thoughts

- We were able to characterize some sets of issues that seem to be related, and surface current directions of work in our community
Final thoughts

- Currently we do this type of discovery by asking one another
  - A method like this offers a path to a list of issues or institutions to contact.
  - Might allow us to catch potential collaborations where communication methods have missed them.
Final thoughts

- The time it takes to review clusters with meaningfully small sets of issues may be prohibitively great.
Final thoughts

- A data set of open issues in our backlogs might in and of itself be helpful to product owners and maybe others to grep against.
Contact

Eliot Jordan
Geospatial Infrastructure Developer, Princeton University
- eliotjordan (github, slack)
Contact

Anna Headley
Digital Infrastructure Developer,
Princeton University

- hackmastera (github)
- hackmaster.a (slack)
Resources

- Our code: https://github.com/hackmastera/samvera-backlogs
- NLTK: https://www.nltk.org
- k-means animation: http://shabal.in/visuals/kmeans/6.html
- Presentation template by SlidesCarnival