RICH CONTEXT CAPSTONE: GRAPH-BASED DATASET RECOMMENDATION SYSTEM

Team Members: Haopeng Huang, Songjian Li, Tanya Nabila, Muci Yu
Mentors: Julia Lane, Jonathan Morgan, Andrew Gordon, and Clayton Hunter

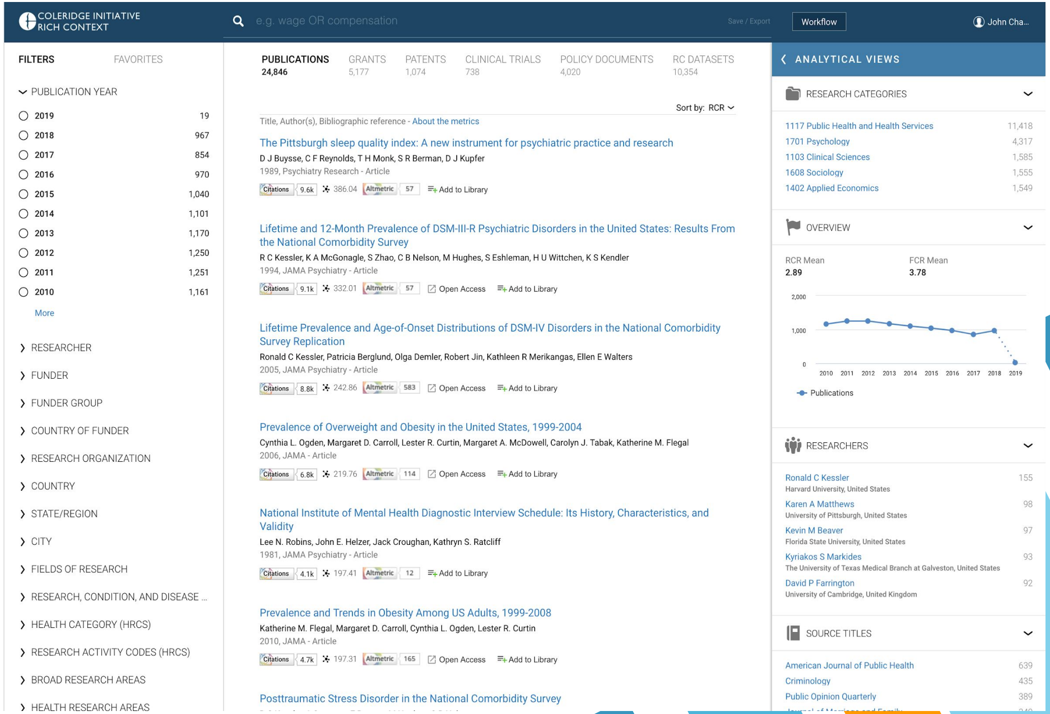
Rich Context Tool (in development)

What?
Dataset & Publication Archive / Search Engine

Purpose?
For a given dataset, find out **who** else worked on the data, on **what topics**, and with **what results**.

To help researchers:

- Increase research efficiency
- Reduce research cost
- Collaborate in related research community



COLERIDGE INITIATIVE

Rich Context Competition

[ ] JSON
{ } 0
{ } 1
publication_id : 102
data_set_id : 312
score : 0.4406929671764373
[ ] mention_list
0 : "Balance Sheet Statistics"
{ } 2
publication_id : 102
data_set_id : 362
score : 0.4406929671764373
[ ] mention_list
0 : "Balance Sheet Statistics"
{ } 3
publication_id : 103
data_set_id : 308
score : 0.5140509486198426
[ ] mention_list
0 : "Securities Holdings Statistics"
{ } 4
publication_id : 103
data_set_id : 314
score : 0.5140509486198426
[ ] mention_list
0 : "Securities Holdings Statistics"

Goal: Automate the discovery of research datasets and the associated research methods and fields in social science research publications.

Input of model: Labeled & unlabeled publication papers from ICPSR Archives

What the competition produced:

» Publication-to-dataset relations
» Publication-to-research methods relations
» Publication-to-research field relations

“Score” indicates the level of confidence in the prediction

Amazon.com for Datasets

Problem Statement

- Great data go undiscovered and are undervalued
- Time and resources are wasted redoing empirical work
- *No existing recommendation system for dataset!*

Our Capstone Goal:

Can we develop an approach to recommend datasets and help improve research efficiency using the relational data we have?

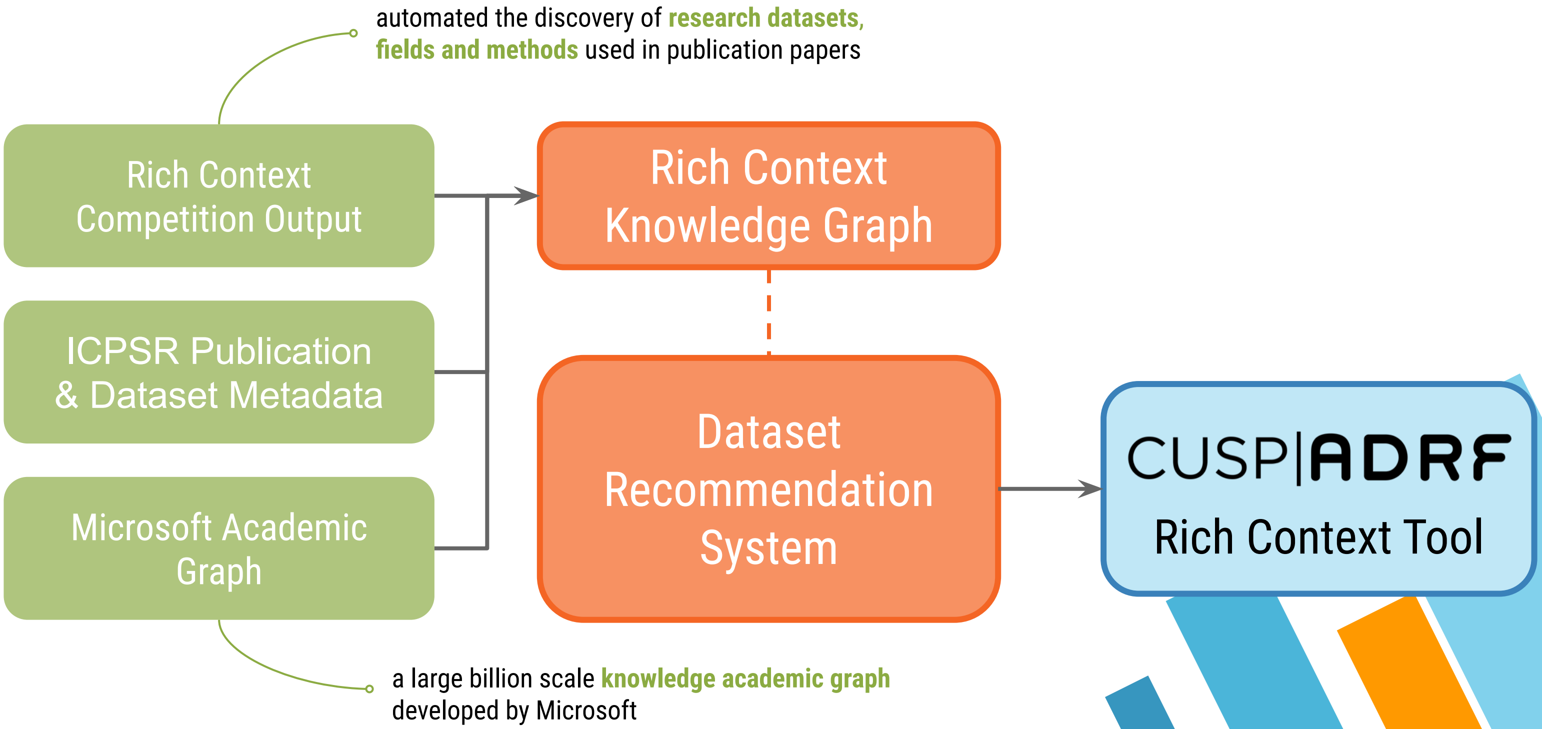
Related to data you've viewed

New data similar to data you've used

What others have done with similar data (recipes)

Recipes like yours

OUR ROLE IN THIS CAPSTONE



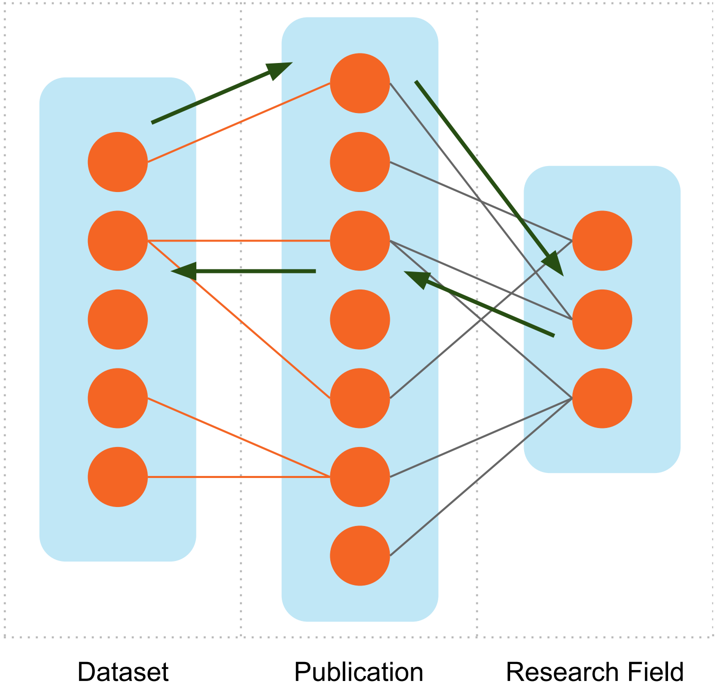
Why Graph-Based Recommendation System?

- Leverages multiple types of entities (datasets, publications, author, keywords, topic)
- Builds a knowledge aware recommendation system
- Scalable & Space Efficient
- Does not require user activity history

Basic Approach:

- Build a heterogeneous knowledge graph network
- Use K nearest nodes to recommend items

Initial Approach (and why it didn't work)



Only used data from rich context competition.

- Measured nearest nodes by path distance & weighted edges
- Sparsely connected dataset-publication connections
- Most of the recommendations given will return the same scores

Revising “nearest nodes” definition

Before: assumed nearest nodes by shortest path & weighted edges

Now: assuming nearest nodes by node similarity algorithms

Simple measurements factors in the common neighbors between the two nodes, whereas more complex ones considers partitioning the network into communities.

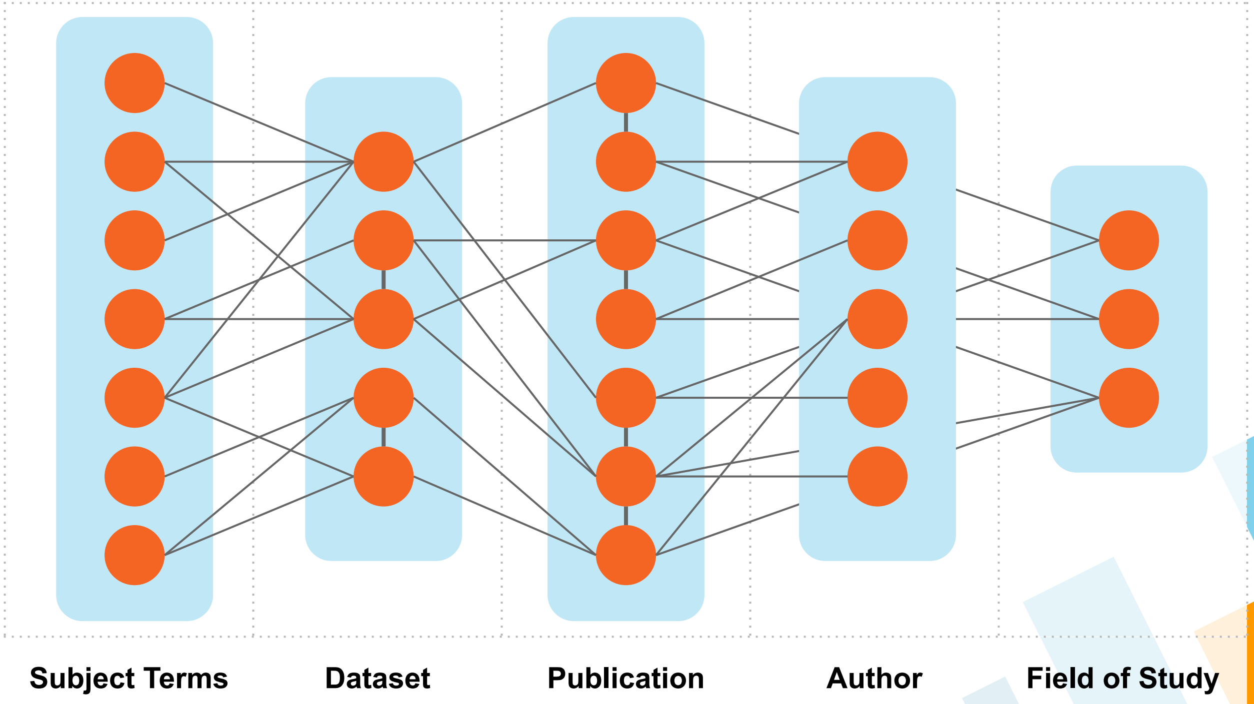
A. Clauset, C. Moore, and M. Newman. Hierarchical structure and the prediction of missing links in networks.

More degrees, more common neighbors, better recommendations.

Revised Approach

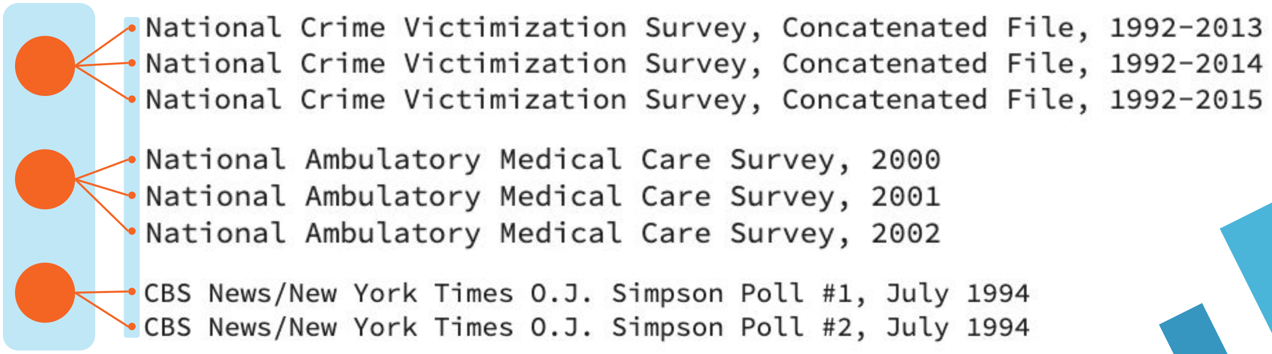
- More connections and entity layers
  - Handling Dataset Versions
  - Add Subject Terms Layer from **ICPSR dataset**
  - Add Publication Similarity Edges using *Word2vec*
  - Add Author & Field of Study Layer from **MAG**
- Apply popular node similarity algorithms
  - *Neighbour-Based* (Jaccard, Cosine, Adamic-Adar)
  - *Community-Based* (Hopcroft)

Rich Context Knowledge Graph

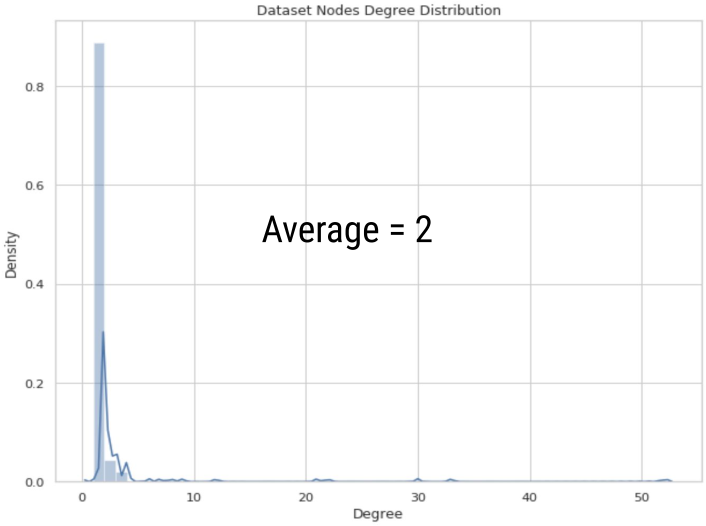


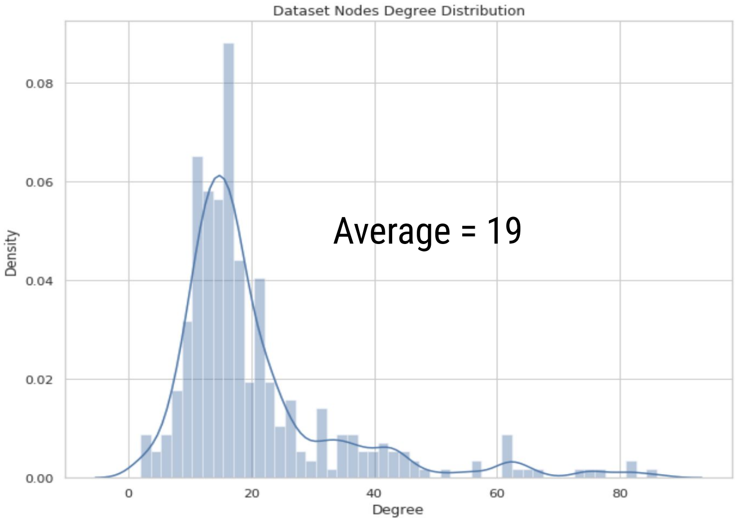
Handling Dataset Versions

- Shrinked Dataset Layer from 10,348 to 6,080 nodes (decreased by 41%)
- Higher average dataset-to-publication connections (increased by 45%)
- Resulting in a more condensed dataset-to-publication relation

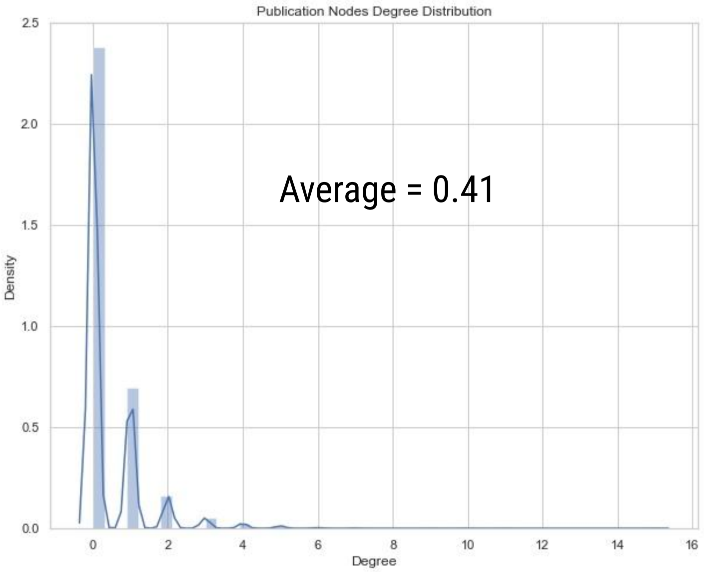


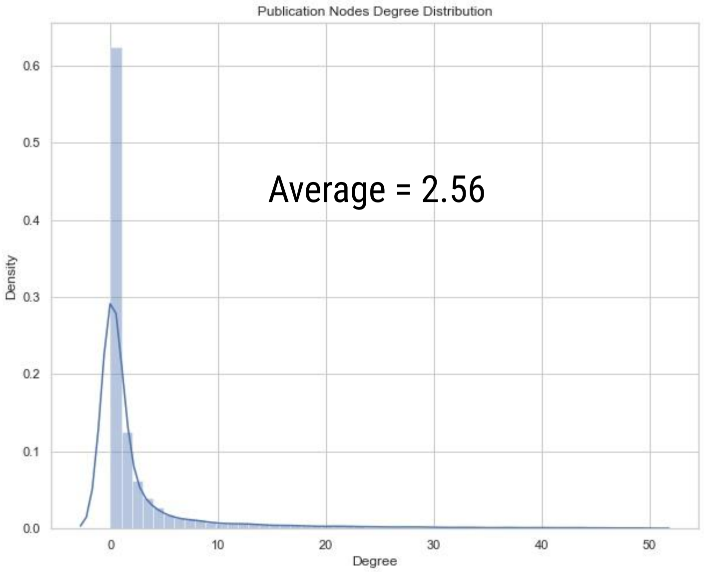
Adding Subject Terms Layer





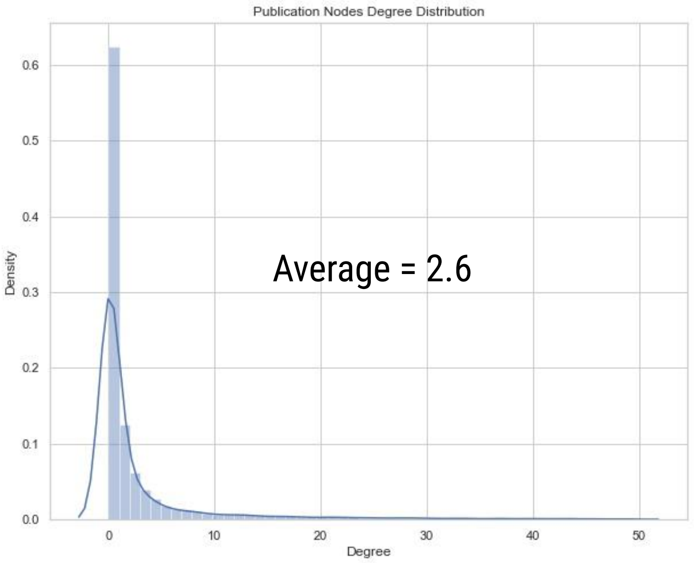
Adding Publication Similarity Connections

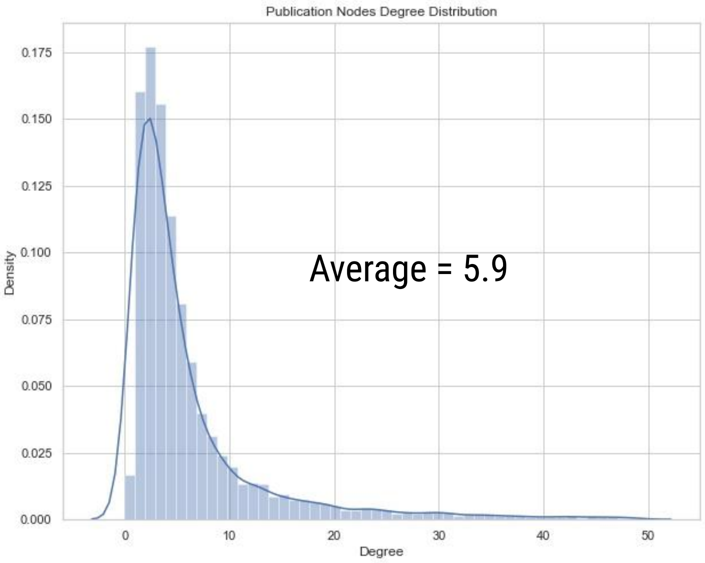




Increased connectivity in publication layer

Adding Author & Field of Study Layers

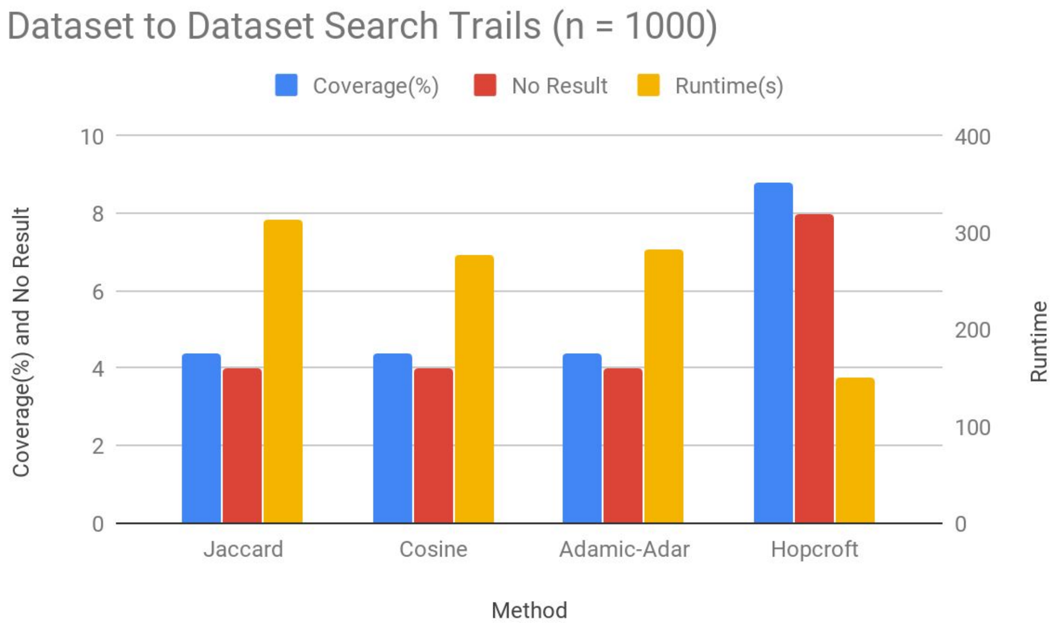




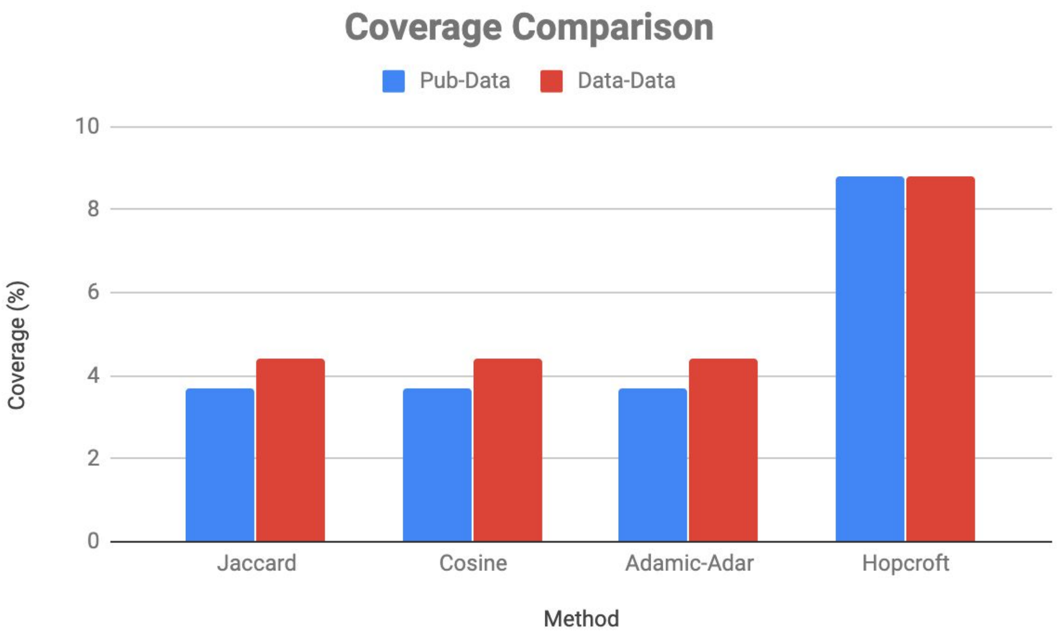
Experimenting with Node Similarity Algorithms

Jaccard	The number of common neighbours is divided by the number of neighbours that exist in at least one of the two objects
Cosine	The number of common neighbours is divided by the total number of possible neighbours
Adamic-Adar	Frequency-weighted common neighbors, implies that rarer features are more telling
Hopcroft	Modification of resource allocation index using community information.

Comparing Node Similarity Algorithms



Comparing Node Similarity Algorithms



Results

- Recommendations from Publication Papers simulations produced less coverage due to lack of connectivity between the publication and dataset layers.
- Hopcroft produced the most coverage and fastest K-nearest nodes due to its community-based calculations.

Future Directions

- Incorporate user feedback for precision & recall evaluations
- Explore more link prediction methods (collaborative filtering, hybrid methods, etc)
- Utilize more of the Microsoft Academic Graph
- Improve dataset-to-publication connectivity

DEMO

Q&A

Comparing Node Similarity Algorithms

Random Simulations: 1000

Algorithm	Coverage(%)	Isolated Nodes	Runtime(s/trail)
Jaccard	37.5	11	0.381
Cosine	39.6	12	0.100
Hopcroft	38.2	0	0.353
Adamic-Adar	3.15	0	0.755

Community Detection: Louvain

The Louvain method of community detection is an algorithm for detecting communities in networks. It maximizes a modularity score for each community, where the modularity quantifies the quality of an assignment of nodes to communities by evaluating how much more densely connected the nodes within a community are, compared to how connected they would be in a random network.

The Louvain algorithm is one of the fastest modularity-based algorithms, and works well with large graphs. It also reveals a hierarchy of communities at different scales, which can be useful for understanding the global functioning of a network.

Link Prediction: Hopcroft

Simple measures consider easy-to-compute factors like the number of neighbors shared between two nodes, whereas more complex definitions partition the network into groups, and then determine the probability that two nodes are connected based on the group memberships of those nodes

Other Node Similarity Measurements

Jaccard: The number of vertices adjacent to both a and b normalized by the number of vertices adjacent to either a or b.

Adamic-Adar: log of sum of the inverses of the degrees of vertices adjacent to both a and b.