NODES 23

Graph-Based Literature Review Tool

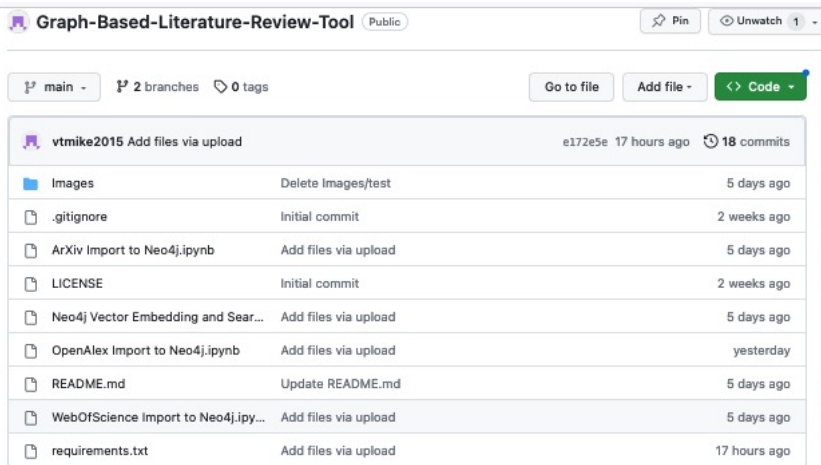# Introduction

Graph Based Literature Review Tool

https://github.com/vtmike2015/
Graph-Based-Literature-Review-Tool

A Network-Graph Based IT Artifact

Abstract

To support theory building, we introduce a network-graph based IT artifact to provide high recall during exploratory searches and high precision using knowledge gained through the literature discovery process. The use of network graphs, where all data is represented as a node, relationship, or property of either, offers a flexible and tailorable methodology able to accommodate the highly iterative process of theory building. This IT artifact was developed to enable aggregation and normalization of data from varied sources and formats to support the acquisition and assessment of literature needed throughout this process. Our goal in presenting this IT artifact is to promote an accessible and pragmatic approach addressing the varied challenges of Information Systems researchers during the information seeking process.

http://hdl.handle.net/10125/80136

NODES 23

What did I want to do?

- Conduct Comprehensive, Systematic Literature Review for PhD Research
- Avoid Implicit Bias Inherent in use any Single Database
- Develop Repeatable Process for Future Research

NODES 23

Where did I start?

Microsoft Academic
Publish or Perish
Google Scholar
Semantic Scholar
Web of Science
arXiV

nodes 23
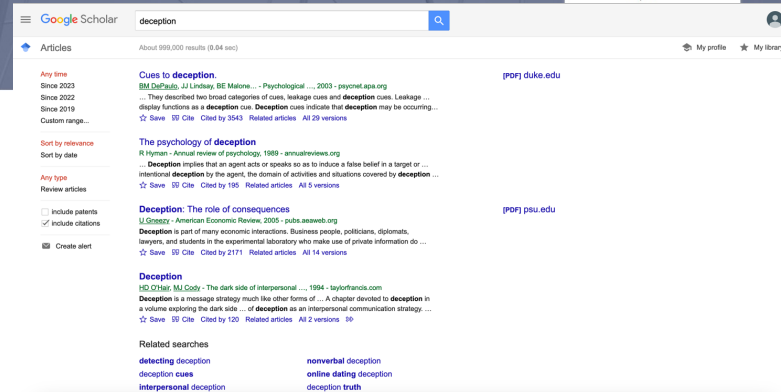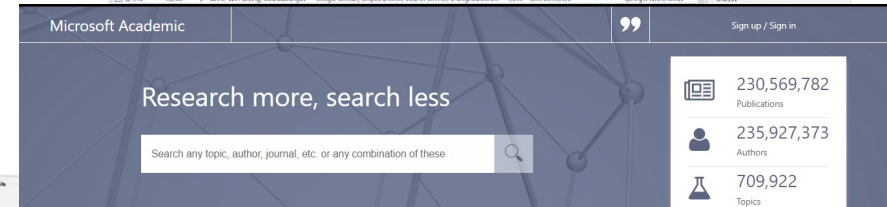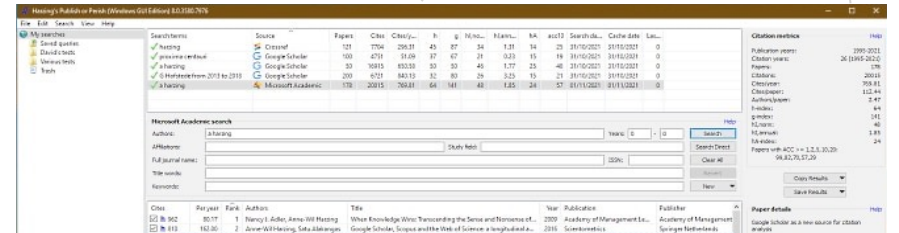
How did I do it in Neo4j?

nodeLabels	propertyName	propertyTypes	OpenAI Mapping	Web Of Science Mapping	arXiv Mapping
["Concept"]	"id"	["String"]	"id"		categories
["Concept"]	"pass"	["Long"]	"pass"		
["Concept"]	"display_name"	["String"]	"display_name"		
["Concept"]	"wikidata"	["String"]	"wikidata"		
["Concept"]	"level"	["Long"]	"level"		
["Concept"]	"score"	["Double"]	"score"		
["Work"]	"source"	["String"]	"OpenAlex"	"Web of Science"	"arXiv"
["Work"]	"id"	["String"]	"id"	UT (Unique ID)	entry_id
["Work"]	"cited_by_api_url"	["String"]	"cited_by_api_url"		
["Work"]	"cited_by_count"	["Long"]	"cited_by_count"	Times Cited, All Databases	
["Work"]	"pass"	["Long"]	"pass"		
["Work"]	"corresponding_author_ids"	["StringArray"]	"corresponding_author_ids"		
["Work"]	"corresponding_institution_ids"	["StringArray"]	"corresponding_institution_ids"		
["Work"]	"created_date"	["String"]	"created_date"		
["Work"]	"display_name"	["String"]	"display_name"	Article Title	title
["Work"]	"doi"	["String"]	"doi"	DOI	doi
["Work"]	"is_paratext"	["Boolean"]	"is_paratext"		
["Work"]	"is_retracted"	["Boolean"]	"is_retracted"		
["Work"]	"language"	["String"]	"language"		
["Work"]	"locations_count"	["Long"]	"locations_count"		
["Work"]	"ngrams_url"	["String"]	"ngrams_url"		
["Work"]	"publication_date"	["String"]	"publication_date"	Publication Date	published
["Work"]	"publication_year"	["Long"]	"publication_year"	Publication Year	
["Work"]	"title"	["String"]	"title"		title
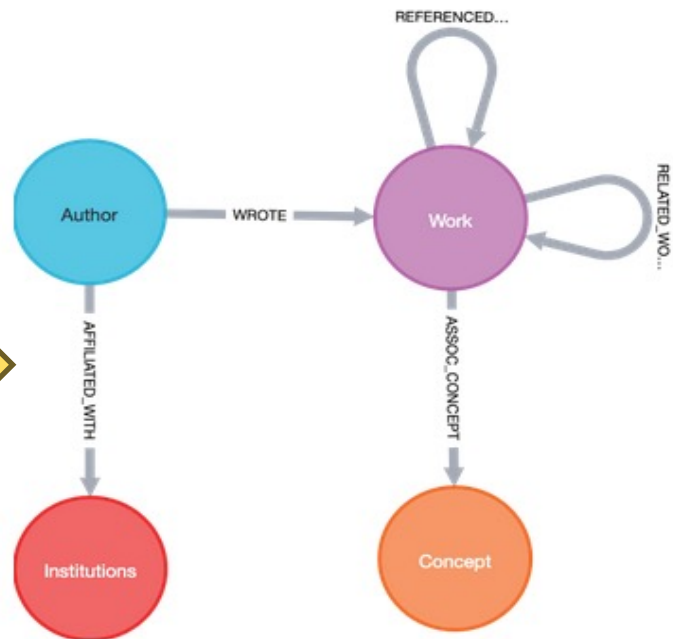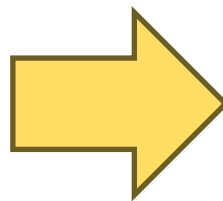
How did I do it in Neo4j? – Grey Literature (Manual)
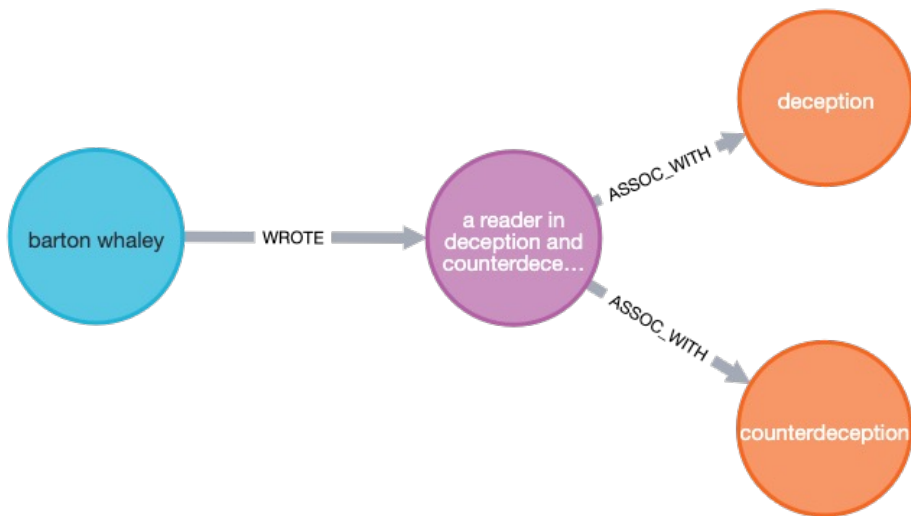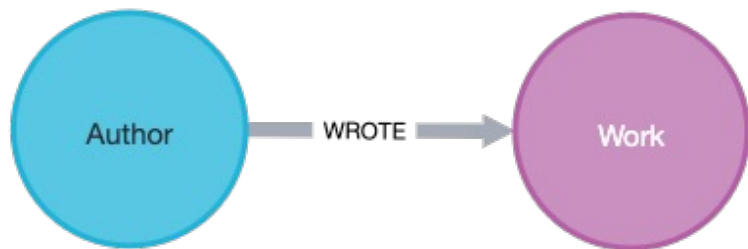

Syntax:

CREATE (w:Work{id: "manual-01", authorships: "barton whaley", display_name: "a reader in deception and counterdeception", type: "unpublished draft", source: "library collection", concepts: ["deception", "counterdeception"]})

MATCH (a:Author),(w:Work) WHERE a.display_name = w.authorships CREATE (a)-[:WROTE]->(w)

NODES 23

How did I do it in Neo4j? – Web of Science (from CSV)

Web of Science
Data Model Schema

Syntax:

"LOAD CSV WITH HEADERS FROM
'file:///wos_file.csv' AS line WITH line WHERE
line.`Authors` IS NOT NULL

MERGE (w:Work {id: line.`UT (Unique ID)`})
SET w.source = 'WebOfScience',
   w.display_name = coalesce(line.`Article Title`, ''),
   w.doi = coalesce(line.`DOI`, ''),
   w.type = coalesce(line.`Document Type`, ''),
   . . .
  w.abstract = coalesce(line.`Abstract`, ''),
  w.authorships = coalesce(split(line.`Authors`, '; '),
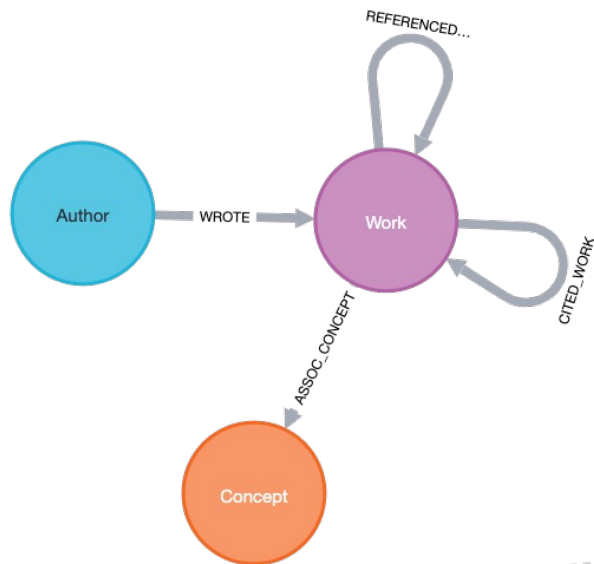 w.ISBN = coalesce(line.`ISBN`, ''),
  w.ISSN = coalesce(line.`ISSN`, '')"

How did I do it in Neo4j? – arXiv & Semantic Scholar (JSON)

arXiV
Data Model Schema
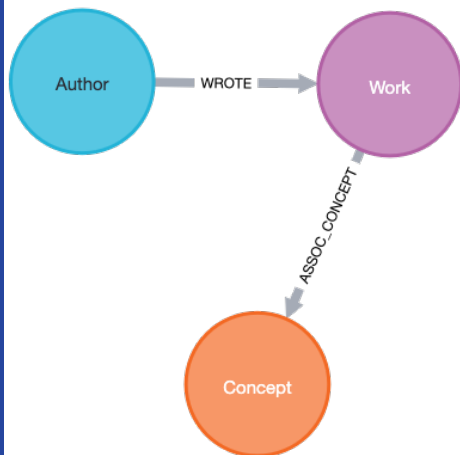
Sematic Scholar
Data Model Schema

Syntax (arXiv):

search = arxiv.Search(
  query = "deception",
  max_results=float('inf'),
  sort_by = arxiv.SortCriterion.SubmittedDate)

"MERGE (w:Work {id: $id}) SET
  w.source = 'arXiv',
  w.display_name = coalesce($display_name,''),
  w.doi = coalesce($doi, ''),
  . . .
  w.abstract = coalesce($abstract, '') ",
  id = result.entry_id,
  display_name = result.title,
  doi = result.doi,
  abstract = result.summary

NODES 23

How did I do it in Neo4j? – OpenAlex (JSON from file)

OpenAlex
Data Model Schema
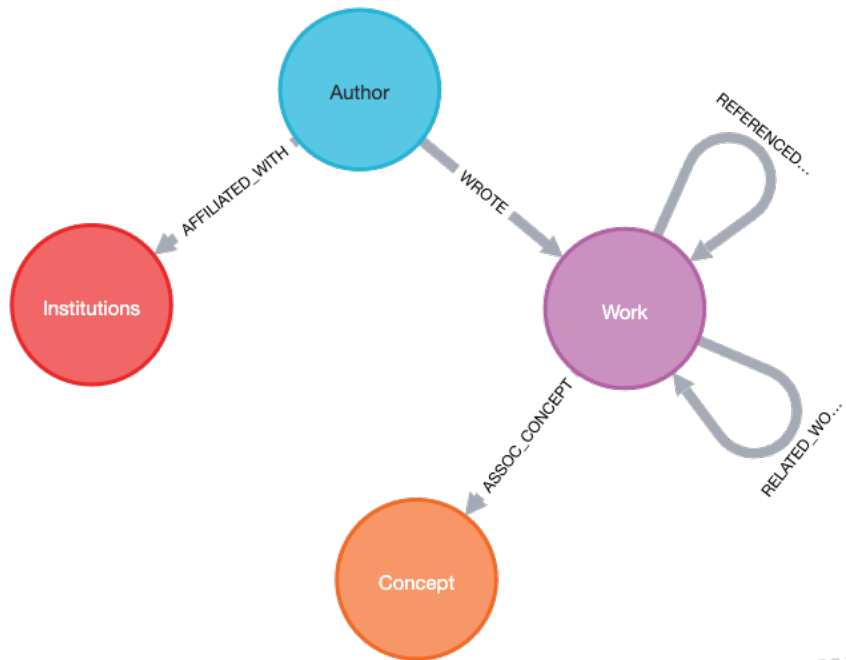
Syntax:

"CALL apoc.periodic.iterate(
"CALL apoc.load.json('works_file.json')
YIELD value","MERGE (w:Work {id: value.id})
  SET w.source = 'OpenAlex',
   w.cited_by_count = coalesce(value.cited_by_count, ''),
   w.corresponding_author_ids =
      coalesce(value.corresponding_author_ids , '')
   w.display_name = coalesce(value.display_name, '')
    …
   CALL apoc.convert.setJsonProperty(w,
   'inverted_abstract', value.abstract_inverted_index) ",
{batchSize: 200, parallel: true, retries: 2}

How did I do it in Neo4j? – All Sources

Graph Search – Theory Ecosystem Comparison

Computer Deception

| Title	Author(s)	Count
Interpersonal Deception Theory	D. Buller, J. Burgoon	165
Testing Interpersonal Deception Theory	D. Buller, J. Burgoon, A. Buslig, J. Roiger	29
Channel Expansion Theory and the Experiential Nature of Media Richness Perceptions	J. Carlson, R. Zmud	22
Toward a general theory of deception	B. Whaley	19
A Survey of Defensive Deception: Approaches Using Game Theory and Machine Learning	M. Zhu et al.	12

Cyber Deception

Title	Author(s)	Count
A Survey of Defensive Deception: Approaches Using Game Theory and Machine Learning	M. Zhu et al.	69
Game theory for adaptive defensive cyber deception	K. Ferguson-Walter, S. Fugate, J. Mauger, M. Major	33
Toward a general theory of deception	B. Whaley	31
Foureye: Defensive Deception Against Advanced Persistent Threats via Hypergame Theory	Z. Wan et al.	19
Game Theory on Attack Graph for Cyber Deception	A. Anwar, C. Kamhoua	18

MATCH p=(w:Work)-[:REFERENCED_WORK]-(n:Work) WHERE w.abstract CONTAINS "cyber" AND w.abstract CONTAINS "deception" AND n.display_name CONTAINS "theory" RETURN DISTINCT(n.display_name), COUNT(n.display_name), COLLECT(w.display_name) ORDER BY COUNT(n.display_name) DESC

NODES 23

Vector Embeddings

Abstract - "Defensive deception is a promising approach for cyber defense. Via defensive deception, a defender can anticipate and prevent attacks . . ."
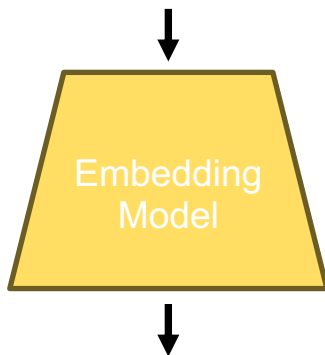
Embedding
Model

Embedding - [-0.029948215931653976,
-0.005470009054988623, 0.011541415005922318,
-0.012460404075682163, -0.012919898144900799,
0.029110314324498177, -0.020433980971574783,
0.004983485676348209, -0.009081769734621048,
-0.0423005037009716, 0.01686614379286766,
0.024691060185432434, -0.020852932706475258,
0.008899323642253876, -0.010730543173849583,
-0.001347230514511466, 0.03457018733024597,
0.004554399289190769, 0.00110565812792629, . . .]

Syntax:

```
record = driver.execute_query (
        "MATCH (w:Work {id: $id}) WITH w
        CALL db.create.setVectorProperty(
            w, 'embedding', $embedding)
        YIELD node RETURN node",

        id = 'https://openalex.org/W3189604664',
        embedding = openai.Embedding.create(
        input = abstract, model =
        "text-embedding-ada002")
            ["data"][0]["embedding"])
```

NODES 23

Vector Search

Work of Interest

"A Survey of Defensive Deception: Approaches Using Game Theory and Machine Learning"

Top 5 Matches (based on abstract)

1. "Game-Theoretic and Machine Learning-based Approaches for Defensive Deception: A Survey" (Score - 0.9968) Note: arXiv pre-print of the Work of Interest
2. "A Game-Theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy" (Score – 0.9678)
3. "Cyber Deception for Computer and Network Security: Survey and Challenges" (Score – 0.9676)
4. "Artificial Intelligence and Game Theory Models for Defending Critical Networks with Cyber Deception" (Score – 0.9666)
5. "Leveraging Computational Intelligence Techniques for Defensive Deception: A Review, Recent Advances, Open Problems and Future Directions" (Score – 0.9665)

```
MATCH (w:Work) WHERE w.id = 'https://openalex.org/W3189604664'
CALL db.index.vector.queryNodes('abstract-embeddings', 55000, w.embedding)
YIELD node AS similarAbstract, score MATCH (n:Work)<-[:WROTE]-(a:Author)
WHERE n.id = similarAbstract.id AND w.id <> similarAbstract.id
RETURN w.id, w.display_name, w.abstract, score, n.id, n.display_name,
COLLECT(a.display_name) AS authors, n.publication_year, n.abstract, n.source LIMIT 5
```

NODES 23

Q&A

- Connect with me on LinkedIn or the event platform to share & learn more.
  - https://www.linkedin.com/in/michael-s-senft/

Why Neo4j?

- Simplicity

- Flexibility

- Scalability

NODES 23