Citation Network

Alisa Leshchenko, Stefan Kolev, Mark Chico, William Darko

CSci 39543 | Fall 2021

Citation Graphs

A citation graph (or citation network), in information science and bibliometrics, is a directed graph that describes the citations within a collection of documents.

Each vertex in the graph represents a document in the collection, and each edge is directed from one document toward another that it cites (or vice-versa depending on the specific implementation). [Wikipedia]

To the right is the Erdős Collaboration graph, a related type of graph where the edges represent co-authorships.

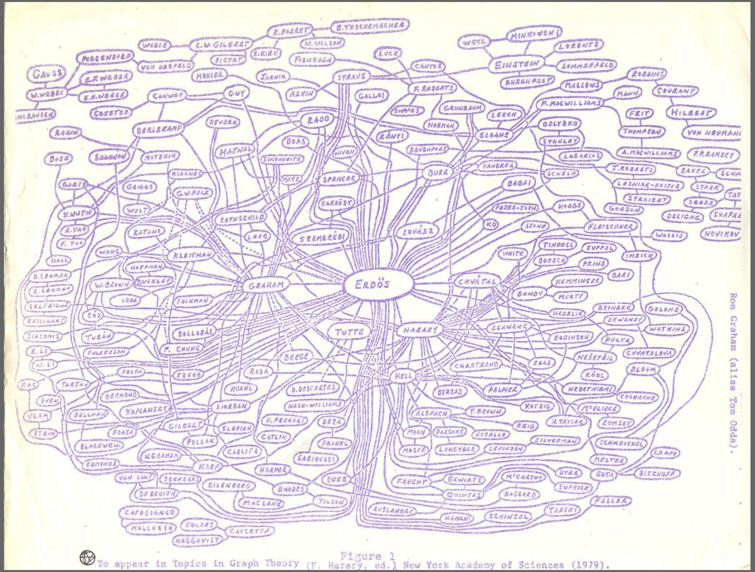


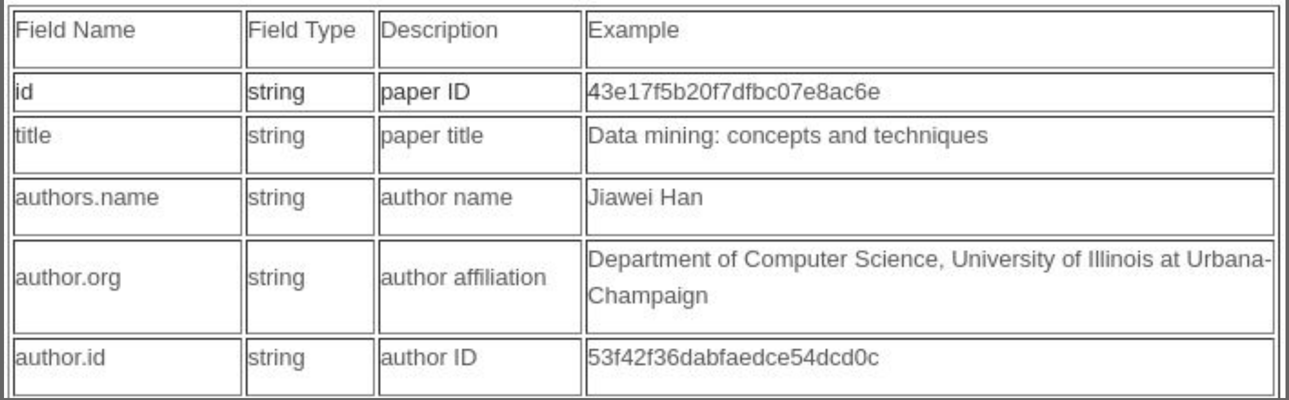
Figure 1

To appear in Topics in Graph Theory (F. Harary, ed.) New York Academy of Sciences (1979).

Ron Graham (alias Tom Odda).

The Data: DBLP V12

The DBLP dataset consists of academic publications extracted from DBLP, ACM, MAG, and other sources. The features of interest for our analysis were: article ID, title, year of publication, fields of study (keywords), and indexed abstract.



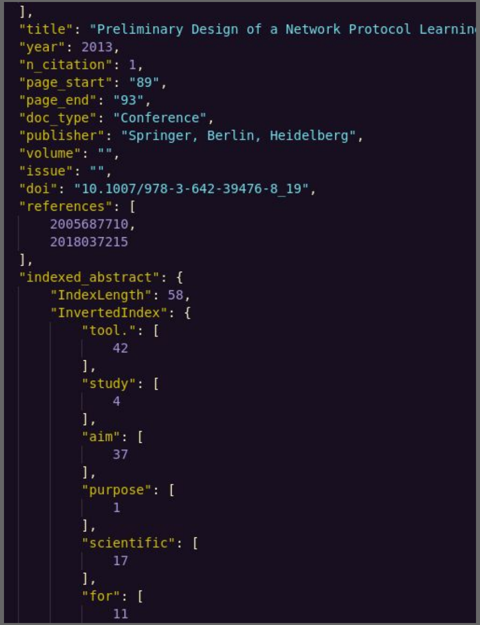
Field Name	Field Type	Description	Example
id	string	paper ID	43e17f5b20f7dfbc07e8ac6e
title	string	paper title	Data mining: concepts and techniques
authors.name	string	author name	Jiawei Han
author.org	string	author affiliation	Department of Computer Science, University of Illinois at Urbana-Champaign
author.id	string	author ID	53f42f36dabfaedce54dcd0c

Data: Format

The JSON file format is very similar to a dictionary. Typically, Python loads a JSON file into memory in its entirety — dictionary-style lookups are then possible. In our case, the data was too large for this workflow.

Notably for our use case, the abstract feature is stored as an “indexed abstract”, and the “fos” feature (field of study) is stored as a dictionary of keywords along with weights.

The indexed abstract included a map where each index has was a word and an array of ints denoting its position throughout the abstract.

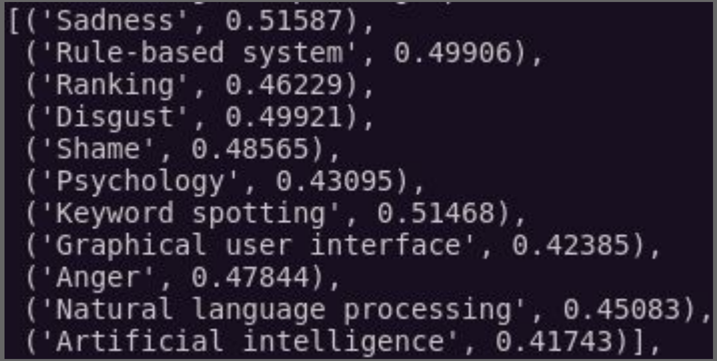


```
],
"title": "Preliminary Design of a Network Protocol Learnin
"year": 2013,
"n_citation": 1,
"page_start": "89",
"page_end": "93",
"doc_type": "Conference",
"publisher": "Springer, Berlin, Heidelberg",
"volume": "",
"issue": "",
"doi": "10.1007/978-3-642-39476-8_19",
"references": [
    2005687710,
    2018037215
],
"indexed_abstract": {
    "IndexLength": 58,
    "InvertedIndex": {
        "tool.": [
            42
        ],
        "study": [
            4
        ],
        "aim": [
            37
        ],
        "purpose": [
            1
        ],
        "scientific": [
            17
        ],
        "for": [
            11
```

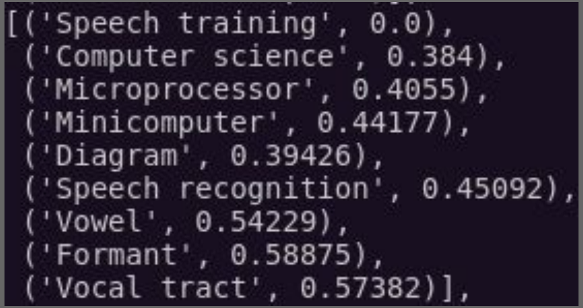
The Data: Challenges

The JSON file is approximately 11.941 GB in size, and must be streamed to be accessed. We ran into difficulties exporting the data into a database, and ultimately had to scrape the file directly, which took a lot of time.

Once we were able to access the data, we found that it was not consistent with the schema on the aminer.org documentation page — the ‘fos’ feature is not a category, but is a set of weighted keywords that is the product of some sort of NLP (i.e. the labels were too noisy and specific to use for training, even taking only the top weight).



```
[('Sadness', 0.51587),
 ('Rule-based system', 0.49906),
 ('Ranking', 0.46229),
 ('Disgust', 0.49921),
 ('Shame', 0.48565),
 ('Psychology', 0.43095),
 ('Keyword spotting', 0.51468),
 ('Graphical user interface', 0.42385),
 ('Anger', 0.47844),
 ('Natural language processing', 0.45083),
 ('Artificial intelligence', 0.41743)],
```



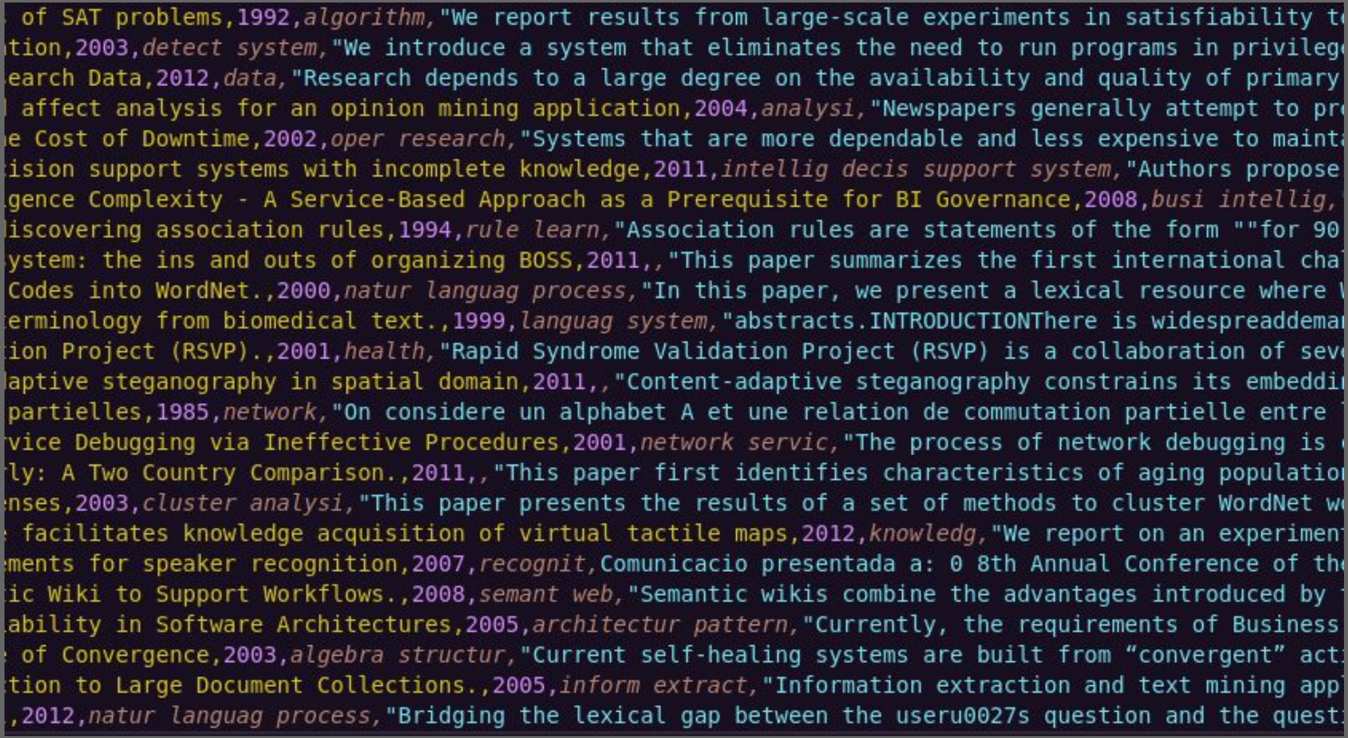
```
[('Speech training', 0.0),
 ('Computer science', 0.384),
 ('Microprocessor', 0.4055),
 ('Minicomputer', 0.44177),
 ('Diagram', 0.39426),
 ('Speech recognition', 0.45092),
 ('Vowel', 0.54229),
 ('Formant', 0.58875),
 ('Vocal tract', 0.57382)],
```

Data: Preprocessing

Our data had to be preprocess in multiple steps. Indexed abstracts were collected and transformed into standard string form for ease of feeding them into a standard NLP preprocessing pipeline, and vectorized using word2vec.

The ‘fos’ feature was transformed using basic NLP in an attempt to wrangle its state space — initially, we had a dictionary of 1,114 categories (taken as the top keyword for every article) for a set of 1,760 articles, which would make classification without overfitting impossible. The final dictionary of categories has 579 words. It was achieved by tokenizing and stemming the keywords, then keeping those tokens that appeared >25 times across all keyword phrases and taking the top keyword if it was not empty, going down to up to the fourth weight in case all higher weights were empty.

Data: After Initial Preprocessing

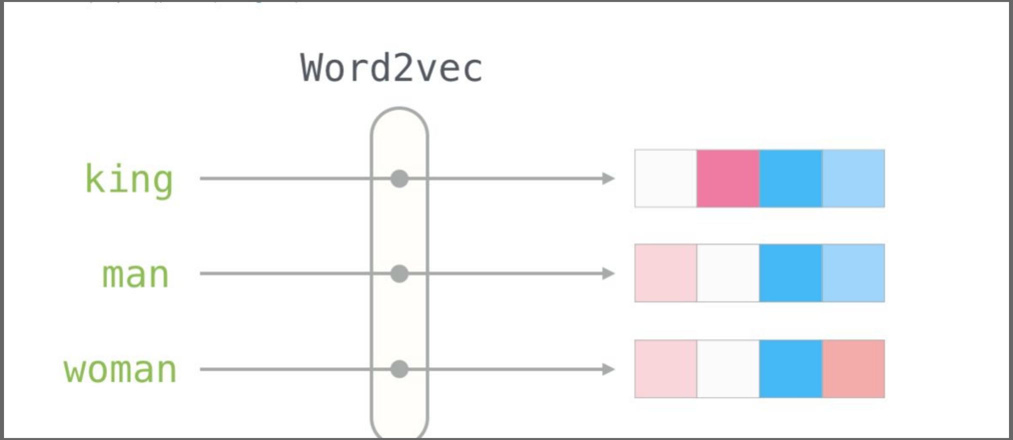


```
 of SAT problems,1992,algorithm,"We report results from large-scale experiments in satisfiability t
tion,2003,detect system,"We introduce a system that eliminates the need to run programs in privileg
earch Data,2012,data,"Research depends to a large degree on the availability and quality of primary
 affect analysis for an opinion mining application,2004,analysi,"Newspapers generally attempt to pr
e Cost of Downtime,2002,oper research,"Systems that are more dependable and less expensive to maint
ision support systems with incomplete knowledge,2011,intellig decis support system,"Authors propose
gence Complexity - A Service-Based Approach as a Prerequisite for BI Governance,2008,busi intellig,
iscovering association rules,1994,rule learn,"Association rules are statements of the form ""for 90
ystem: the ins and outs of organizing BOSS,2011,,"This paper summarizes the first international cha
Codes into WordNet.,2000,natur languag process,"In this paper, we present a lexical resource where
erminology from biomedical text.,1999,languag system,"abstracts.INTRODUCTIONThere is widespreaddema
ion Project (RSVP).,2001,health,"Rapid Syndrome Validation Project (RSVP) is a collaboration of sev
aptive steganography in spatial domain,2011,,"Content-adaptive steganography constrains its embeddi
partielles,1985,network,"On considere un alphabet A et une relation de commutation partielle entre
vice Debugging via Ineffective Procedures,2001,network servic,"The process of network debugging is
ly: A Two Country Comparison.,2011,,"This paper first identifies characteristics of aging populatio
nses,2003,cluster analysi,"This paper presents the results of a set of methods to cluster WordNet w
 facilitates knowledge acquisition of virtual tactile maps,2012,knowledg,"We report on an experimen
ments for speaker recognition,2007,recognit,Comunicacio presentada a: 0 8th Annual Conference of th
ic Wiki to Support Workflows.,2008,semant web,"Semantic wikis combine the advantages introduced by
ability in Software Architectures,2005,architectur pattern,"Currently, the requirements of Business
 of Convergence,2003,algebra structur,"Current self-healing systems are built from “convergent” act
tion to Large Document Collections.,2005,inform extract,"Information extraction and text mining app
,2012,natur languag process,"Bridging the lexical gap between the useru0027s question and the quest
```

Vectorizing Abstracts & Word Embeddings

After transforming the indexed abstracts into a standard string, we used a Word2Vec model to vectorise each abstract. The end result is a word embedding, typically a real-valued vector that encodes the "meaning" of each word, in the context of other words in the vector space.

Vectors closer to each other in the embeddings space are expected to have similar meanings, for example words, and their synonyms.



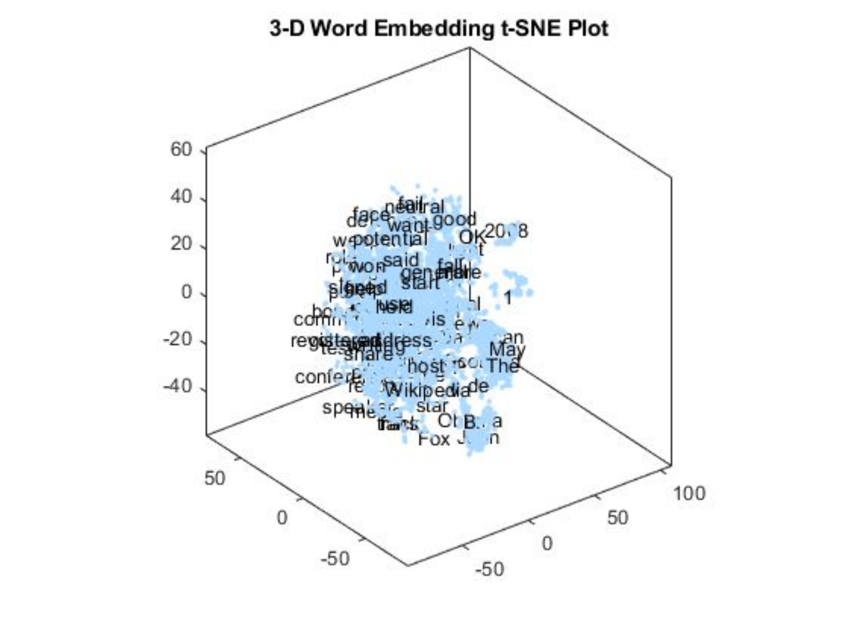
Word2Vec

Word2Vec is an NLP technique published in 2013 by a team of researchers at Google. It’s implemented as a group of related two-layer neural network models, trained to reconstruct linguistic contexts of words.

Taking as an input a large corpus of text, the Word2Vec model produces a vector space, typically of several hundred dimensions.

Word2Vec utilizes either Continuous Bag-of-Words (CBOW) or Skip-Gram, along with hierarchical softmax or negative sampling.

CBOW allows the extrapolation of one word from the context and Skip-Gram allows the extrapolation of a context from one word.



Neo4j Graph

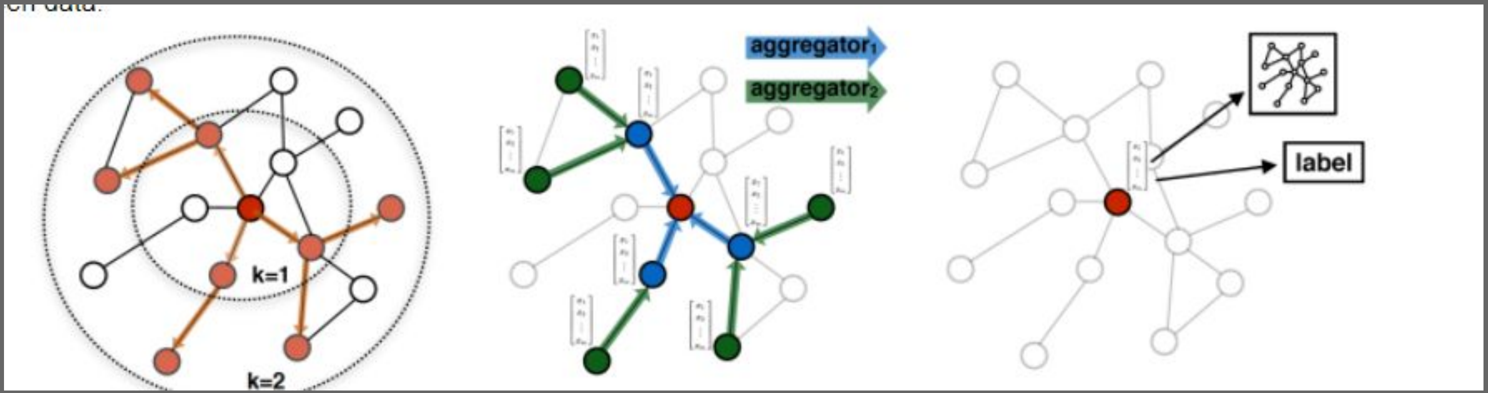
Graph databases like Neo4j allow for the efficient manipulation of highly relational data (data in which rows refer to one another).

We imported the citation data into Neo4j using papers as nodes and citations as edges. Each node contains the id, abstract, and keywords associated with the respective paper. Abstracts were first vectorized using word2vec to help with graph similarity analysis.

Dimensionality Reduction: GraphSAGE

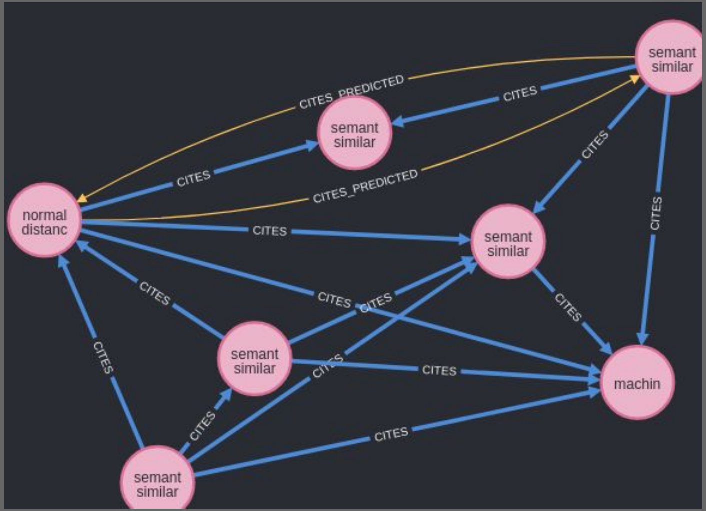
GraphSAGE is a framework applied on graphs that generates low-dimensional vector representations of each node in the graph which generates efficient embeddings for a graph.

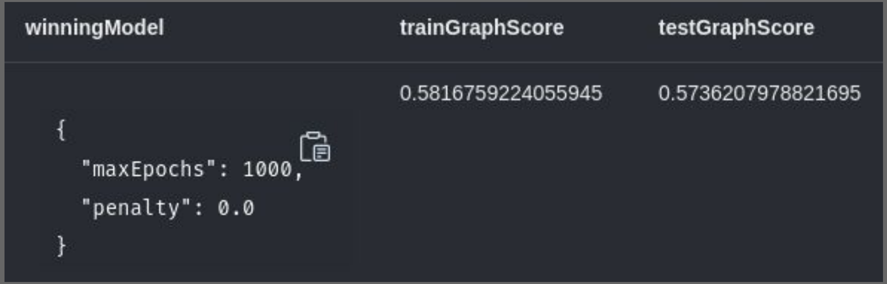
It works especially well on attribute rich nodes that can improve training. It aggregates neighboring nodes and creates group embeddings.



Link Prediction

Link prediction uses the topology and properties of a graph to predict new relationships between existing nodes. In our case, the semantics of a predicted link are indirect: we are predicting which authors are likely to cite other authors in the future, using papers as a proxy. Neo4j offers several variants of link prediction natively.





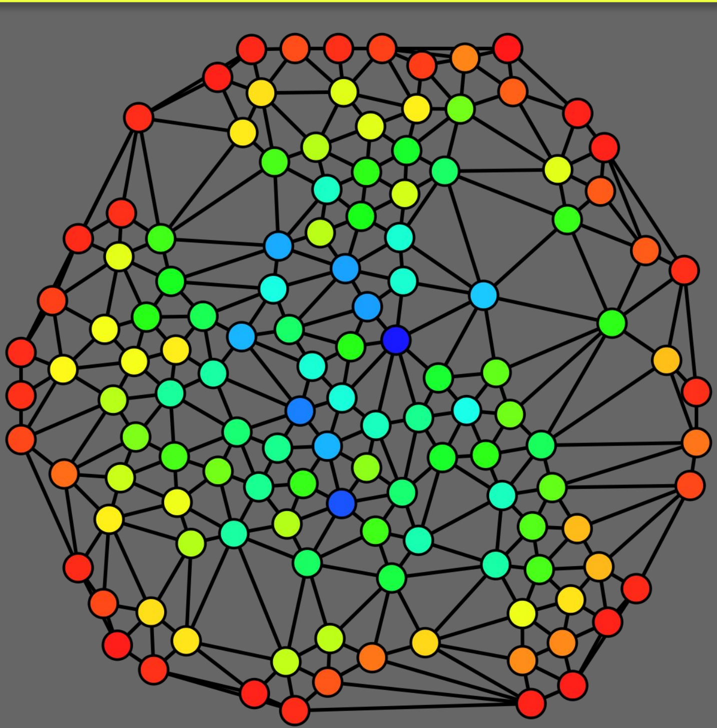
winningModel	trainGraphScore	testGraphScore

{
"maxEpochs": 1000,
"penalty": 0.0
}
	0.5816759224055945	0.5736207978821695

Betweenness Centrality

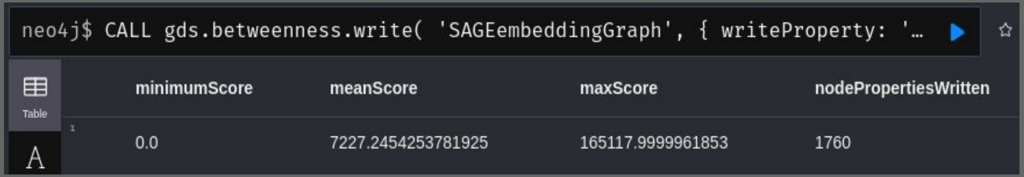
Betweenness Centrality an algorithm used in detecting the amount of influence a node has over the “flow” of information in a graph. It is often used in searching for nodes that serve as a bridge from one part of the graph to another.

The algorithm calculates unweighted shortest paths between all pairs of nodes, with each node receiving a score based on the number of shortest paths passing through it. Nodes that frequently between shortest paths between other nodes, will have higher Betweenness Centrality scores.



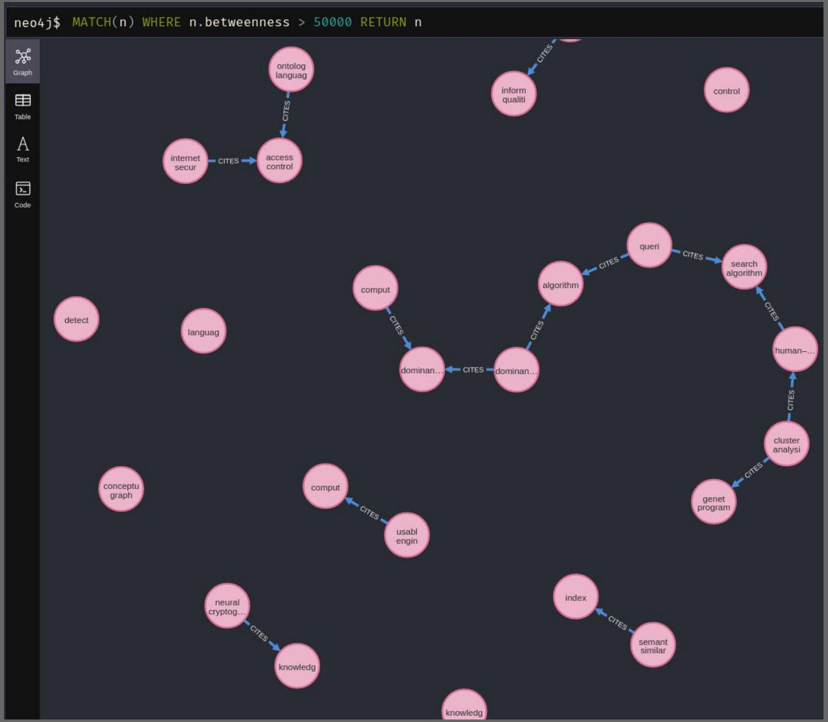
Betweenness Centrality

Displayed here are papers having a betweenness centrality score of over 50,000. These are papers that form influential links across disparate community clusters (i.e. they either cite or are cited by papers in a broad range of subjects).



```
neo4j$ CALL gds.betweenness.write( 'SAGEembeddingGraph', { writeProperty: '…
```

	minimumScore	meanScore	maxScore	nodePropertiesWritten
1	0.0	7227.2454253781925	165117.9999961853	1760



```
neo4j$ MATCH(n) WHERE n.betweenness > 50000 RETURN n
```

Community Detection (The Louvain Method)

The Louvain Method is a community detection algorithm typically used for large networks. More specifically, the Louvain is a hierarchical clustering algorithm, that recursively merges communities into a single node.

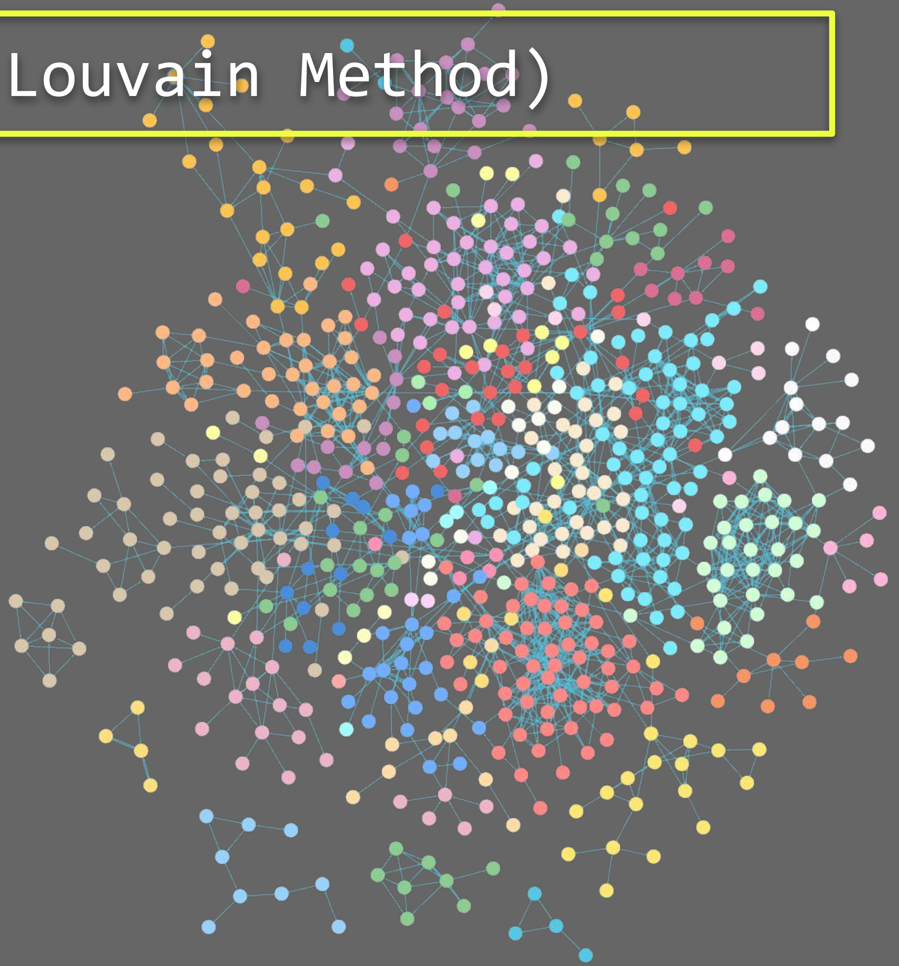
It maximizes a modularity score for each community, where the modularity quantifies the quality of assignment of nodes to communities. This evaluates how much more densely connected the nodes within a community are, compared to how connected they would be in a random network.

For our use case, the Louvain Method can help identify communities of nodes (in our case papers), with similar properties, beyond citations.

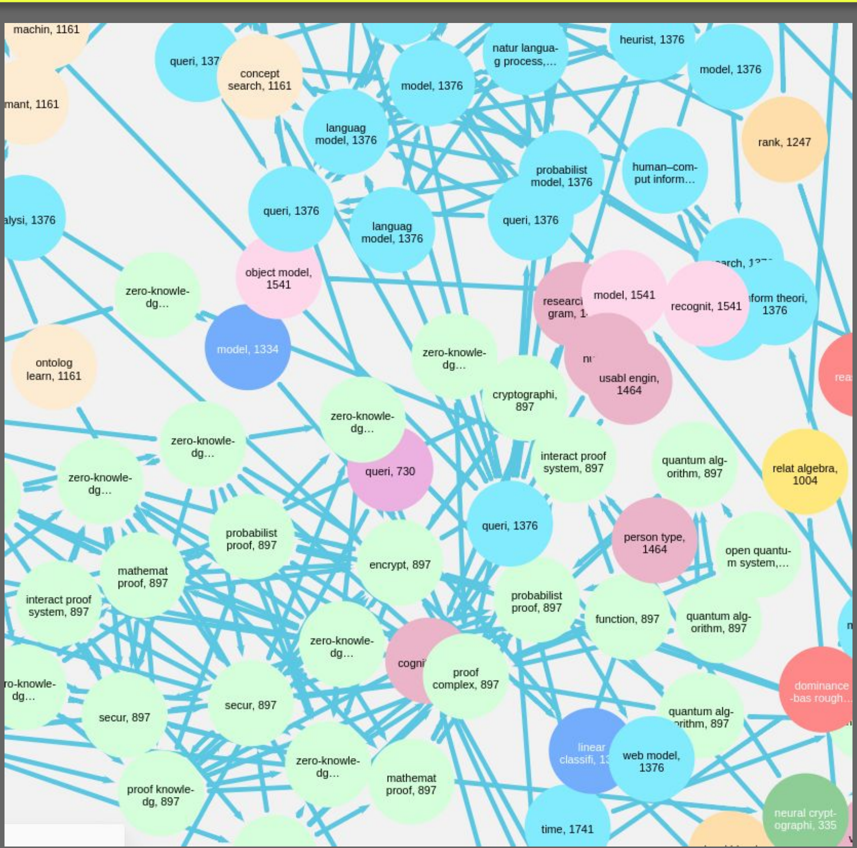
Community Detection (The Louvain Method)

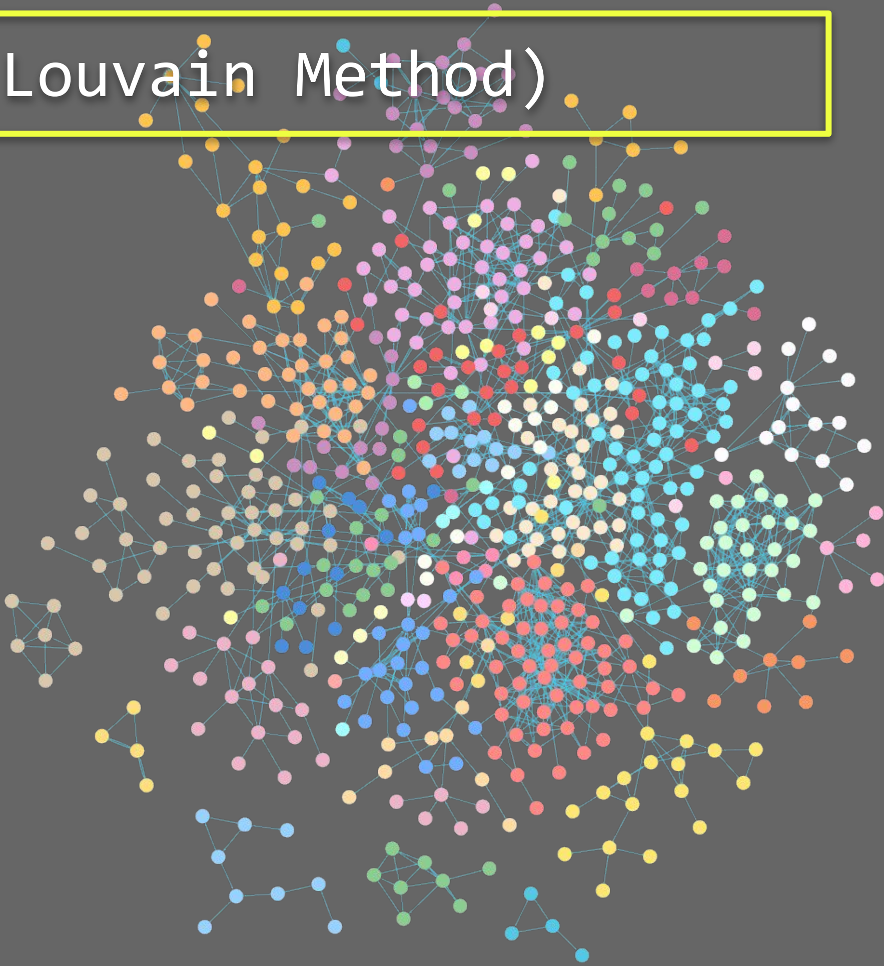
Our data displays a robust community pattern, which is unsurprising because of the way it was extracted: starting with the first 200 papers in the JSON and collecting all papers these cited.

What is interesting are the links across communities.



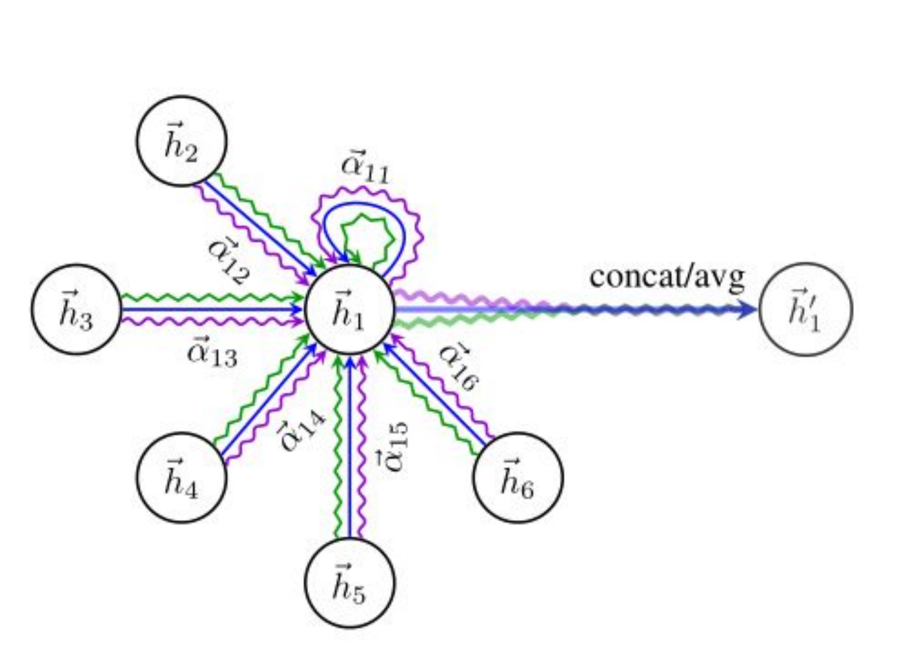
Community Detection (The Louvain Method)





Graph Attention Network

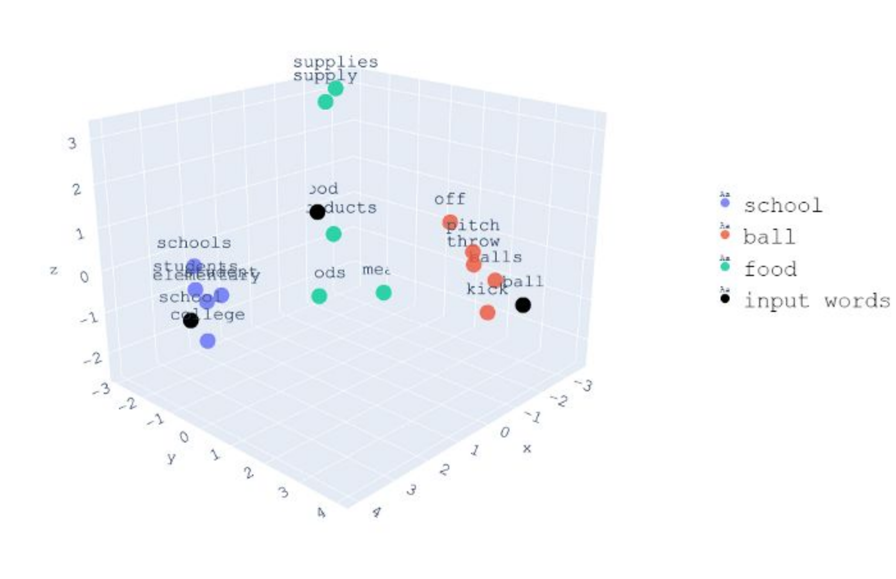
Performs node classification by taking the features of each node, and the adjacency matrix, as the input for a weight matrix that produces feature embeddings. It uses a attention layer which masks out the influence of less relevant nodes on the output based on the adjacency matrix, for each node. Missing features are then estimated.



Principal Component Analysis

An unsupervised feature extraction technique. It is a dimensionality reduction algorithm that can classify nodes using the embeddings we get from the abstract. Lowering the dimensionality of our data also allows it to be represented visually

Example(Not Our Model)

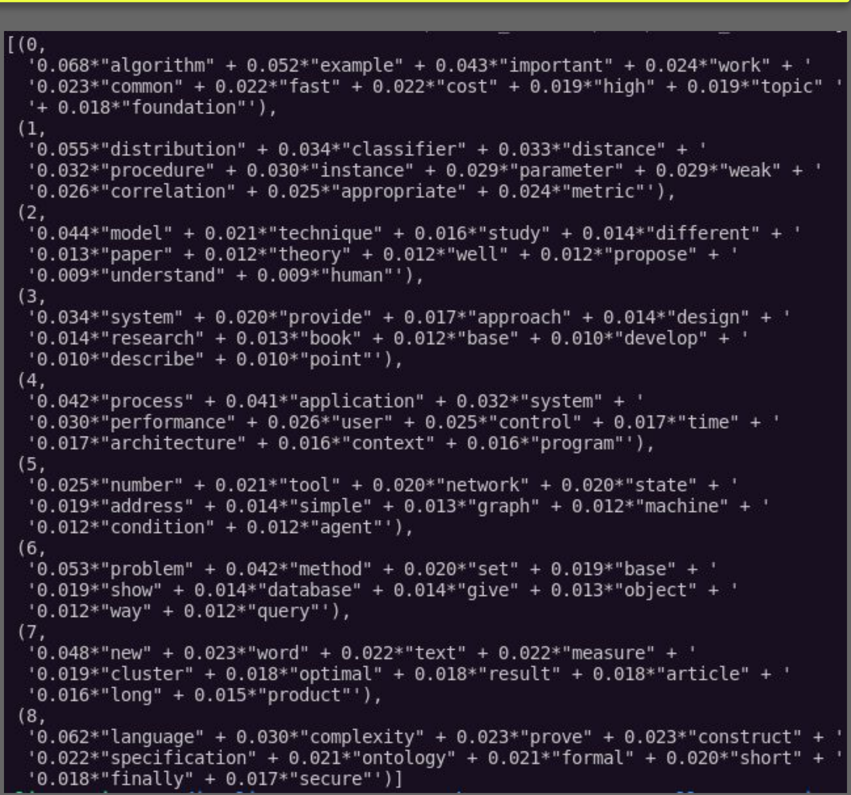


Other Models: SVM and LDA

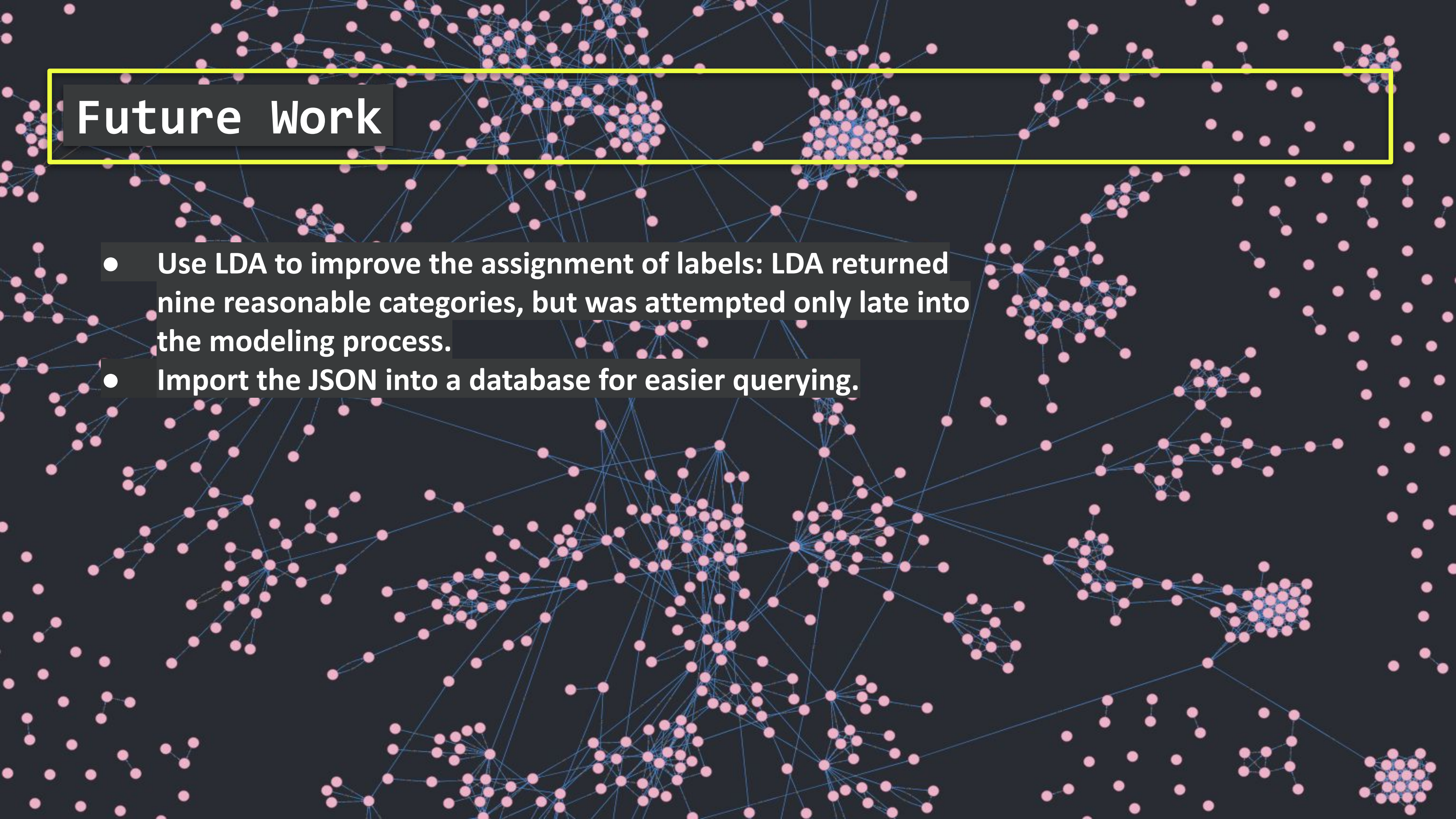
- We expected that the quality of the labels would make classification difficult, and this was confirmed when we ran a linear SVM on the data, with 'fos' label as the target:

Mean accuracy training on vectorized abstract feature: 0.01955
Mean accuracy training on SAGE embedding: 0.00955

- LDA was run on the raw abstracts and was able to pick out reasonable topics. We suspect this is because there were clusters of fairly different texts.



```
[(0,
  '0.068*"algorithm" + 0.052*"example" + 0.043*"important" + 0.024*"work" + '
  '0.023*"common" + 0.022*"fast" + 0.022*"cost" + 0.019*"high" + 0.019*"topic" '
  '+ 0.018*"foundation"'),
 (1,
  '0.055*"distribution" + 0.034*"classifier" + 0.033*"distance" + '
  '0.032*"procedure" + 0.030*"instance" + 0.029*"parameter" + 0.029*"weak" + '
  '0.026*"correlation" + 0.025*"appropriate" + 0.024*"metric"'),
 (2,
  '0.044*"model" + 0.021*"technique" + 0.016*"study" + 0.014*"different" + '
  '0.013*"paper" + 0.012*"theory" + 0.012*"well" + 0.012*"propose" + '
  '0.009*"understand" + 0.009*"human"'),
 (3,
  '0.034*"system" + 0.020*"provide" + 0.017*"approach" + 0.014*"design" + '
  '0.014*"research" + 0.013*"book" + 0.012*"base" + 0.010*"develop" + '
  '0.010*"describe" + 0.010*"point"'),
 (4,
  '0.042*"process" + 0.041*"application" + 0.032*"system" + '
  '0.030*"performance" + 0.026*"user" + 0.025*"control" + 0.017*"time" + '
  '0.017*"architecture" + 0.016*"context" + 0.016*"program"'),
 (5,
  '0.025*"number" + 0.021*"tool" + 0.020*"network" + 0.020*"state" + '
  '0.019*"address" + 0.014*"simple" + 0.013*"graph" + 0.012*"machine" + '
  '0.012*"condition" + 0.012*"agent"'),
 (6,
  '0.053*"problem" + 0.042*"method" + 0.020*"set" + 0.019*"base" + '
  '0.019*"show" + 0.014*"database" + 0.014*"give" + 0.013*"object" + '
  '0.012*"way" + 0.012*"query"'),
 (7,
  '0.048*"new" + 0.023*"word" + 0.022*"text" + 0.022*"measure" + '
  '0.019*"cluster" + 0.018*"optimal" + 0.018*"result" + 0.018*"article" + '
  '0.016*"long" + 0.015*"product"'),
 (8,
  '0.062*"language" + 0.030*"complexity" + 0.023*"prove" + 0.023*"construct" + '
  '0.022*"specification" + 0.021*"ontology" + 0.021*"formal" + 0.020*"short" + '
  '0.018*"finally" + 0.017*"secure"')]
```



Future Work

- Use LDA to improve the assignment of labels: LDA returned nine reasonable categories, but was attempted only late into the modeling process.
- Import the JSON into a database for easier querying.