Rich Context Algorithms & Development: Dataset Recommendation System

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

Github Repository: https://github.com/rich-context-capstone-2019/Rich-Context-Capstone.git
Website: https://rich-context-capstone-2019.github.io/Rich-Context-Capstone/

Abstract

Exploring research datasets and their relational network between one and the others, and between datasets and entities of interest, such as research fields, paper titles, authors, and citation counts, are currently inefficient and lack an integrated online platform that improves this process. This capstone project connects datasets with various entities in the papers that use them and with other related papers. It improves user experience by building a dataset recommendation system based on a graph model. We create this system based on model outputs from the Rich Context Competition[1] which showed weak mean relationship scores between datasets, publications, and research fields. We developed a novel evaluation metric to assess the performance of our system based on our what our definition of a good dataset recommendation system. Our previous network version adding keywords entity shows better connectivity based on a similarity matrix that prioritizes the shortest path between two datasets. And our latest knowledge network remapping fields of study entity and adding author entity shows even better connectivity. After we apply different connections of nodes and weighted edges in our network and define different similarity matrices, we produce a network model with the best performance and build our recommendation system on it. This recommendation system can be very useful when integrated with an interactive search engine and lead researches in all domains.

Introduction

Knowledge about datasets is often embedded with those analysts who use specific datasets frequently or data stewards who are responsible for managing and giving access to the data. At the same time, there are many papers and reports - including those in published journals or in the "grey literature" at different agencies or organizations - that use a mix of datasets. The ADRF team is developing a Rich Context platform that will leverage a mix of text analysis, machine learning, and user friendly search interface to improve the knowledge dissemination and contribution ecosystem. The newly ended Rich Context Competition, a competition held by Coleridge Initiative to develop content extraction models from publication papers and extract the set of datasets used and mentioned in the papers as well as their research fields, has explored model and system approaches by teams and institutions to initially test the feasibility and accuracy of such a tool which provides our team examples and directions to improve and

[1] http://coleridgeinitiative.org/richcontextcompetition

further explore the application of the platform[2] [Hunter & Lane, 2019].[3] Other existing resources include an initial search tool that is also from *coleridgeinitiative.org* and the Dimensions platform from the website which is publicly available for non-commercial use without the dataset addition. Lastly, a Rich Context evaluation tool from the CUSP ADRF team is built as an initial tool for people to manually provide mentions of datasets in publications.

This project investigates how to best leverage the heterogeneous information, such as entity relations and textual content, and use it to generate dataset recommendations for our users. Among different recommendation techniques, we propose a novel approach to integrate a graph based recommendation system which combines techniques from our empirical studies and literature reviews.

Related Works

Recommender systems can be helpful in the digital system library to provide users with relevant products and information by predicting a user's interest in a product based on various types of information such as a user's profile, past purchases, product features and their interactions. Traditional recommendation systems developments have mainly focused on item-to-item similarity for content-based approach or user-to-user similarity for collaborative filtering approach. Since collaborative filtering methods usually suffer from cold-start problems, many works attempt to leverage additional information to improve recommendation performance [Chuan Shi, 2017]. Recent developments show attempts to test the idea of using a graph-based recommender system that naturally combines the content-based and collaborative approaches [Zan Huang et al. 2002]. However, it is important to note that past studies regarding scientific paper or research-related content recommendation systems have reported that for such algorithms, *no gold standard exists against which new systems can be evaluated* [Laura Steinert et al 2016].

Our research looks at the study of scientific paper recommendation systems using a heterogeneous network (or knowledge graph), since this is more closely relevant compared to a user-to-product recommendation systems approach. Incorporating knowledge graph into recommender system seems promising in improving the recommendations and provide explainability in the process [Y. Cao 2019]. Many graph based approach was proposed to exploit the information of connectivity within a knowledge graph and alleviate the data sparsity problem that exist within the digital library. Utilizing a knowledge graph to make recommendations also has its many approaches, one of them being using a link prediction to conduct a hybrid of user and item based recommendation [Zan Huang 2005].

We also looked into the most comprehensive and currently developed open sourced academic knowledge network, the Microsoft Academic Graph. The information provided can vary in all forms of academic knowledge today: journal articles, conference proceedings, books, patents, datasets (codes), etc. Given such a highly connected graph, we extracted some of these information and approach to eventually incorporate some into our network.

[2] https://coleridgeinitiative.org/assets/docs/adrf_documentation_whitepaper.pdf

[3] CUSP Spring 2019 Capstone Projects Overview

Problem Statement

In searching for and assessing different types of recommender systems and their mechanisms, we know the importance of testing and finding a best-fit graph-based similarity measure for our own system. The publication-to-entities recommendation generation would require different similarity measures and network structure from that of the publication-to-publication scenario. Furthermore, limited user profile and interaction data lead us to learn session-based recommendation approaches and design a customized version given the conditions of our project which can be algorithmically and managerially challenging.

Our project goal is to improve research efficiency and user experience on connecting publication papers with relevant datasets by generating dataset recommendation to users based on, for instance, a research field or their most recent search activity. The foundation of the system will be based on graph model which captures relationships between datasets and multiple types of entities such as publications, research fields, authors, and so forth. This approach of integrating entities and relationships among them into a unique graph structure is intended to design recommendations as graph traversals. This will in turn offer a flexible framework which can handle various entities of interest and enable developers to compute recommendations for use cases of all kinds.

The prototype of a working recommendation system model has been built to generate recommendations based on a specified set of user interaction. These interactions include searching for the most relevant dataset based on a publication paper or dataset the user has recently viewed or searched upon based on a keyword. Another interaction also include entering a keyword that matches one of our subject terms and field of study database and retrieve the most relevant dataset from there. The relevancy from one node to another will be determined by a working link prediction algorithm which has been experimented and evaluated in this project.

Data

Our project aims to utilize the Rich Context Competition results first, then gather external sources of additional information to improve our graph network. The current three main sources of our dataset are the Rich Context Competition results for publication-to-dataset, the ICPSR archive for the publication and dataset metadata, and lastly the Microsoft Academic Graph for building our rich context knowledge graph.

Rich Context Competition Dataset
The Rich Context Competition held by Coleridge Initiative was with the intent to automate the discovery of research datasets, fields, and methods used in a publication paper. This project have used both the dataset used to train the model as well as the prediction output from the winner of the competition. The datasets used include 3 main components: publication-to-dataset pairs, publication-to-research field pairs, and publication-to-dataset mention pairs. Each contain a "score" property, indicating a probability score on a scale of 0 to 1 representing the level of confidence that the publication is in the stated research field. These datasets are seen in Appendix, Figure 1-2.

ICPSR Archive

The ICPSR Archive is considered an International Leader in Data Stewardship, maintaining a data archive of about 250,000 of research in the social and behavioral sciences. The ICPSR Bibliography of Data-related Literature is a searchable database that as of 2019 contains 80,000 citations of published and unpublished works resulting from analyses of data held in the ICPSR archive. This project is using about 15,000 publication papers and 10,348 dataset from the ICPSR archive in which we have extracted their metadata, such as subject terms, descriptions, titles, and DOI.

Microsoft Academic Graph Data

The Microsoft Academic Graph (MAG) currently has 218,951,661 papers with 239,743,363 authors, 664,149 topics, 4,384 conferences, 48,731 journals, and 25,511 institutions[4]. Since we extracted the author entity data from the graph first, here we introduce how the author entity data was originally collected and cleaned by the MAG team. They collected author data from two types of sources: (1) feeds from publishers (e.g. ACM and IEEE), and (2) web-pages indexed by Bing [Sinha et al., 2015].

Currently we are using the MAG dataset to extract Authors and Field of Study of all the publications we currently have in our dataset. We have mapped our publication papers to MAG's publication papers by their DOI. The final publication-author pairs data we make that is ready to be added into our network is with 48,157 new edges. There are 39,355 unique Author ID and 14,481 unique papers connected to our existing ICPSR dataset. The publication-FOS pairs data we added into our network is with 131,540 new edges. There are 15,242 unique Field of Study ID and 14,479 unique papers connected to our existing ICPSR dataset. The process of getting the final publication-author pairs and publication-FOS pairs data out of the original accessible data from MAG is discussed in the Methodology section below.

Methodology

After comparing various methods for building the recommendation system and assessing the risks and limitations of each approach, we decided to pursue the network graph-based recommendation. Our first prototype network consists of three types of entity layers; datasets, publications, and research fields. This first approach did not work well because our nodes were so sparsely connected that our dataset nodes received mostly the same scores for recommendation. Based on our network analysis and further research, we sought to develop our network on top of the additional data we have collected.

Network Development

We've built our graph progressively, and evaluating the results upon each iteration. In order to handle the sparsity of connections in our dataset layer, we integrated subject terms of the datasets. In order to add richer context in publication paper layer, we've added authors and field of study layer into our network. By adding the additional layers, the degrees of publication and dataset nodes increased by a significant amount. That means our new network is more connected, shows more common neighbors, and will provide more dispersed scores for the recommendations. Figure 2 shows the comparison of the original network graph and the networks with additional layers, and Figure 3 shows the improvement of the degrees of dataset nodes. Our next step is to integrate these two network graphs into one graph so that we can further improve the connectivity of the entities.

[4] https://www.openacademic.ai/oag/

Handling Dataset Versioning

After analysing our dataset nodes, we have noticed that a lot of these datasets are of the same title, but have different versioning. Some of these datasets have the exact same title and method of collection, but different version or cohort. We stripped down numbers and roman numerals in the titles and contracted the same datasets into a single node. Therefore, any recommended items will disregard any versioning of the dataset and focus on the content. This way, the connections in our network regarding the dataset layer are expected to be more condensed and insightful.

Publication-Author Edges (using MAG)

To access the MAG data, we first created a Microsoft Azure Storage account to receive automated distribution of new versions of MAG.[5] Next we created an Azure Databricks service to process the MAG data with an enormous size subscribed in our Azure storage and used Python to wrangle and preprocess our data without downloading it.[6] Then we used our set of publications to extract Paper ID, Author ID, Display Name, Paper Count and other attributes that only match the doi's in our publication data. We got two useful datasets that include author data and paper data from the MAG. Once we matched all the Publication ID (our data) with Paper ID (MAG) we further added the Author ID (MAG) and finalized the Publication-Author Pairs file which was ready to be added into our network.

Publication to Fields-of-Study Edges (using MAG)

Similar to the process above, we used Azure Databricks notebooks to process the MAG data online. We used our set of publications to extract Paper ID, Field of Study ID, Normalized Name, and other attributes that only match the doi's in our publication data. Once we matched all the Publication ID (our data) with Paper ID (MAG) we further added the Field of Study ID (MAG) and finalized the Publication-FOS Pairs file which was ready to be added into our network.

Publication Similarity (using Word2Vec)

To improve the connectivity of our graph, we have added edges among publications with their similarity scores as weights, so that datasets used by similar publications will receive higher ranking in recommendation. We calculated document similarities based on word embedding and cosine similarity. First, we preprocess the text files --- tokenizing, removing stop words, stemming and vectorizing the words using the pre-trained Google News Word2Vec model. In the Word2Vec model, words with similar meaning will be represented by similar vectors. We take the average of all word vectors in a publication and use the result as the vector representation of the document. Then we calculate the cosine similarity between every pair of publications. In our database, there are 5000 publications, so there are more than 12 millions pairs of publications. The similarity score ranges from 0.40 to 0.99 with the mean around 0.85.

Link Prediction As Node Similarity Measurements

Our node similarity measures in the first prototype phase was defined as the shortest path distance from one node to another. After further research on measuring node similarity measures in our heterogeneous graph network, we decided to redefine our approach by using link prediction algorithms normally find in knowledge graphs and social network graphs [Symeonidis et al].

[5] https://docs.microsoft.com/en-us/academic-services/graph/get-started-setup-provisioning

[6] https://docs.microsoft.com/zh-cn/academic-services/graph/get-started-setup-databricks

The link prediction algorithms used in this project are Jaccard coefficient, cosine distance, Hopcroft and Adamic-Adar index. The first two of the link prediction algorithms are calculations based on common neighbors, commonly used in social network graphs [Liben-Nowell 2003]. The Jaccard coefficient is a commonly used metric in information retrieval, which measures the probability that both node *x* and *y* have a feature *f* (which in this case to be the neighbors of the graph). The Adamic Adar conducts a related measure, a simple approach of counting the common features by weighting rarer features more heavily. Both Hopcroft and Adamic-Adar index is a resource allocation index calculated from all node pairs using the community information. Here in this project, we are using the Louvain community detection approach. These newly defined node similarity calculations will play a major role in determining the ranking of the most relevant set of datasets in our recommendation system since we are doing so by choosing the K-nearest neighbor from any given node.

Evaluation Methods

We used offline evaluation to assess the performance of our system as well as other measurements that act as a proxy to assess how well the output of our recommendation system is. The first evaluation metric is on how *diverse* can our recommendation system give its recommended sets. This measures how dissimilar the recommended items are for a user. The second evaluation metric is the *coverage* of our recommendations within the entirety of our dataset collection. This measures the percentage of items the recommender system will be able to recommend. The third will be on having a human to use our system and evaluate the sample sets of recommended items themselves. In the future, after the system is deployed, the users can refine our system by pruning the list of suggested papers with a positive or negative feedback by declaring a subset relevant or irrelevant. Other important evaluations and analysis regarding our methods will involve basic network analysis techniques such as taking into account the density, the degree distributions, the connectivity, the runtime of the algorithms, and the amount of node isolations detected in the graph.

Results & Analysis

We have developed our network and algorithmic approach gradually while analysing its effects between each step. The phases of our development can be divided into three phases, as shown in Figure 3 (see Appendix). The first phase of the network have used solely data from the rich context competition and built three layers of entity: dataset, publication, and research field. The second phase, we've added 2 additional layers, still from the competition output, the subject terms connected to the dataset layer and the research field connected to the publications. The third and final phase of the network development has shown great strides since we found ways to better connect the graph, such as adding publication similarity edges, adding an author layer, and switching the research field layer to field of study from the MAG dataset. In Figure 4, the majority of the datasets nodes in our previous network graph has less than 3 degrees, and the majority of datasets nodes' degree has been increased to 11 in our updated network. It means that our network graph are now more connected and are able to produce a more diverse set of relevant nodes. With a more connected network graph and accurately computed similarity scores, we can make recommendations with more coverage and diversity.

The effect of adding Subject Terms Layer

Based on our network analysis and node similarity approach, the main reason our recommendation was not working as well was due to lack of connectivity between the Publication Paper layer and the Dataset layer. A total of 12,457 dataset citations was found in these publications, yet this only involved around five thousand unique publication and 1,646 unique dataset. There were 10,348 dataset nodes which meant that our network only has a 16% coverage of all datasets and the rest was not connected to any publication. To increase the dataset coverage, we connect dataset node with another entity layer, the subject terms of each dataset. An average of 19 subject terms can be found in a dataset node, and this distribution can be seen in Figure 8. The effect of this can be shown in Figure 4, where the dataset degree distribution has increased significantly after adding the subject terms layer.

The effect of adding Publication Similarity by Word2Vec

Another attempt to alleviate the data sparsity problem between the dataset and publication layer, is to conduct a document similarity algorithm to connect similar publication nodes. This way, two similar publications can share a link and have a more connected graph. We have added edges for those with a similarity score of higher than 0.95, which means that 20,364 new edges has been made and connected. The degree distribution of the publication layer has increased from around 5 to around 100 (Figure 7).

The effect of contracting dataset

Treating different versions of the same dataset as one dataset node has successfully made our network density improve significantly, from 0.0010 to 0.38979. This is due to the shrinkage of the dataset layer from a total of unique 10,348 dataset nodes that have decreased to 58% of its size, down to a more compact and dense dataset layer of 6,080 nodes.

The effect of adding Author Layer & Field of Study Layer

To get a richer dimension in terms of publication-to-publication associations, we've decided to add an extra layer, that is the Author layer. This will help to alleviate the sparsity that exist in the publication layer where previously only 26% of our publications were successfully connected in the graph while the rest suffered from isolation. As seen in Figure 7, the degree distributions of the publication layer has increased significantly. This will in turn produce more dispersed node similarity scores and a more connected publication layer.

Conclusions

We built up the graph-based dataset recommendation system to improve research efficiency. The advantages of graph-based recommendation system include that it can leverage multiple types of entities and information, and it can generate results without requiring user activity history. We have utilized the available resources and improved upon our previous prototype network. Our preliminary network analysis has shown that we have successfully created a more connected network with deeper layers of entities, which results in more diverse sets of connections. After successful efforts in creating more connected heterogeneous networks, we experiment on the network graphs and start to produce recommendation rankings. We conducted multiple recommendation ranking approaches and evaluated the results based on our predefined evaluation methods. The algorithms we used to calculate nodes similarity include Jaccard similarity, Cosine similarity, Hopcroft algorithm, and Adamic-Adar Index. Due to our evaluation metrics, the Cosine similarity algorithm has the best performance in terms of coverage and runtime. The evaluation metrics for similarity algorithms are shown in Appendix, Figure 9.

Our future work should focus on using user feedback to dynamically update our recommendation system. For now, our evaluation is limited to objective performance metrics such as diversity, coverage and computational efficiency. Qualitative evaluation on individual recommendation results is important, but will only be available when the system is online and we start to track user data. Particularly, we can use these data to put more weight on the edges connecting relevant nodes. This will allow the system to generate more customized recommendation based on user preference.

Team's collaboration statement

The team has split the workload equally amongst all members.

References

Cai, Xiaoyan, Junwei Han, Shirui Pan, and Libin Yang. "Heterogeneous Information Network Embedding Based Personalized Query-Focused Astronomy Reference Paper Recommendation." *International Journal of Computational Intelligence Systems* 11, no. 1 (2018): 591. doi:10.2991/ijcis.11.1.44.

Huang, Zan, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. "A Graph-based Recommender System for Digital Library." *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL '02*, 2002. doi:10.1145/544229.544231.

Kessler, M. M. "Bibliographic Coupling between Scientific Papers." *American Documentation* 14, no. 1 (01 1963): 10-25. doi:10.1002/asi.5090140103.

Ludewig, Malte, and Dietmar Jannach. "Evaluation of Session-based Recommendation Algorithms." *User Modeling and User-Adapted Interaction* 28, no. 4-5 (10, 2018): 331-90. doi:10.1007/s11257-018-9209-6.

Mcnee, Sean M., Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. "On the Recommending of Citations for Research Papers." *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work - CSCW '02*, 2002. doi:10.1145/587078.587096.

Quadrana, Massimo, Paolo Cremonesi, and Dietmar Jannach. "Sequence-Aware Recommender Systems." *ACM Computing Surveys* 51, no. 4 (07, 2018): 1-36. doi:10.1145/3190616.

Shi, Chuan, and Philip S. Yu. "Recommendation with Heterogeneous Information." *Heterogeneous Information Network Analysis and Applications Data Analytics*, 2017, 97-141. doi:10.1007/978-3-319-56212-4_5.

Huang, Zan, et al. “Link Prediction Approach to Collaborative Filtering.” *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL '05*, 2005, doi:10.1145/1065385.1065415.

Small, Henry. "Co-citation in the Scientific Literature: A New Measure of the Relationship between Two

Documents." *Journal of the American Society for Information Science* 24, no. 4 (07 1973): 265-69. doi:10.1002/asi.4630240406.

Steinert, Laura, and H. Ulrich Hoppe. "A Comparative Analysis of Network-Based Similarity Measures for Scientific Paper Recommendations." *2016 Third European Network Intelligence Conference (ENIC)*, 09 2016. doi:10.1109/enic.2016.011.

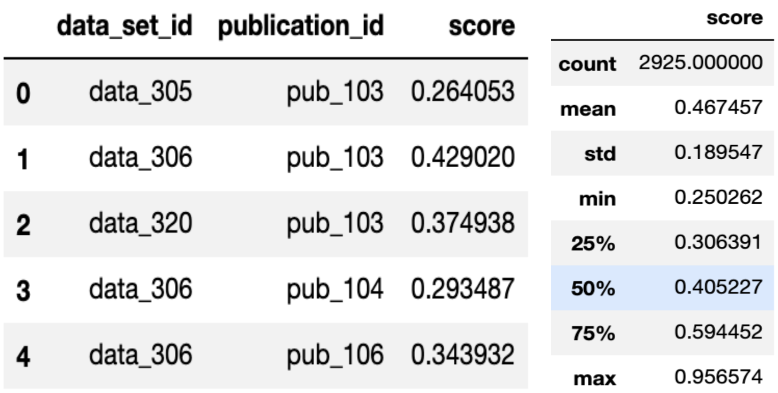
Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion). ACM, New York, NY, USA, 243-246. DOI=http://dx.doi.org/10.1145/2740908.2742839

Liben-Nowell, David, and Jon Kleinberg. "The Link Prediction Problem for Social Networks." *Proceedings of the Twelfth International Conference on Information and Knowledge Management - CIKM '03*, 2003, doi:10.1145/956958.956972.

Cao, Yixin, et al. "Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences." The World Wide Web Conference on - WWW '19, 2019, doi:10.1145/3308558.3313705.

Symeonidis, Panagiotis, et al. "Transitive Node Similarity for Link Prediction in Social Networks with Positive and Negative Links." Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys '10, 2010, doi:10.1145/1864708.1864744.

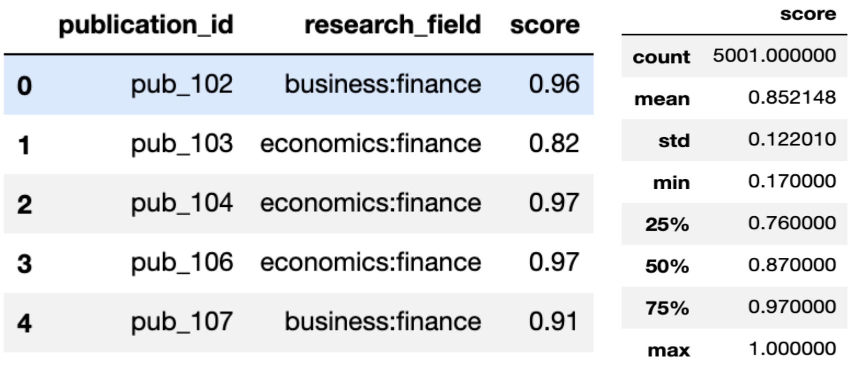
Appendix



	data_set_id	publication_id	score
0	data_305	pub_103	0.264053
1	data_306	pub_103	0.429020
2	data_320	pub_103	0.374938
3	data_306	pub_104	0.293487
4	data_306	pub_106	0.343932

	score
count	2925.000000
mean	0.467457
std	0.189547
min	0.250262
25%	0.306391
50%	0.405227
75%	0.594452
max	0.956574

Figure 1: the relationship between data set and publication id. The "score" being the confidence score of the edge prediction.



	publication_id	research_field	score
0	pub_102	business:finance	0.96
1	pub_103	economics:finance	0.82
2	pub_104	economics:finance	0.97
3	pub_106	economics:finance	0.97
4	pub_107	business:finance	0.91

	score
count	5001.000000
mean	0.852148
std	0.122010
min	0.170000
25%	0.760000
50%	0.870000
75%	0.970000
max	1.000000

Figure 2: the relationship between data set and publication id. The "score" being the confidence score of the edge prediction.

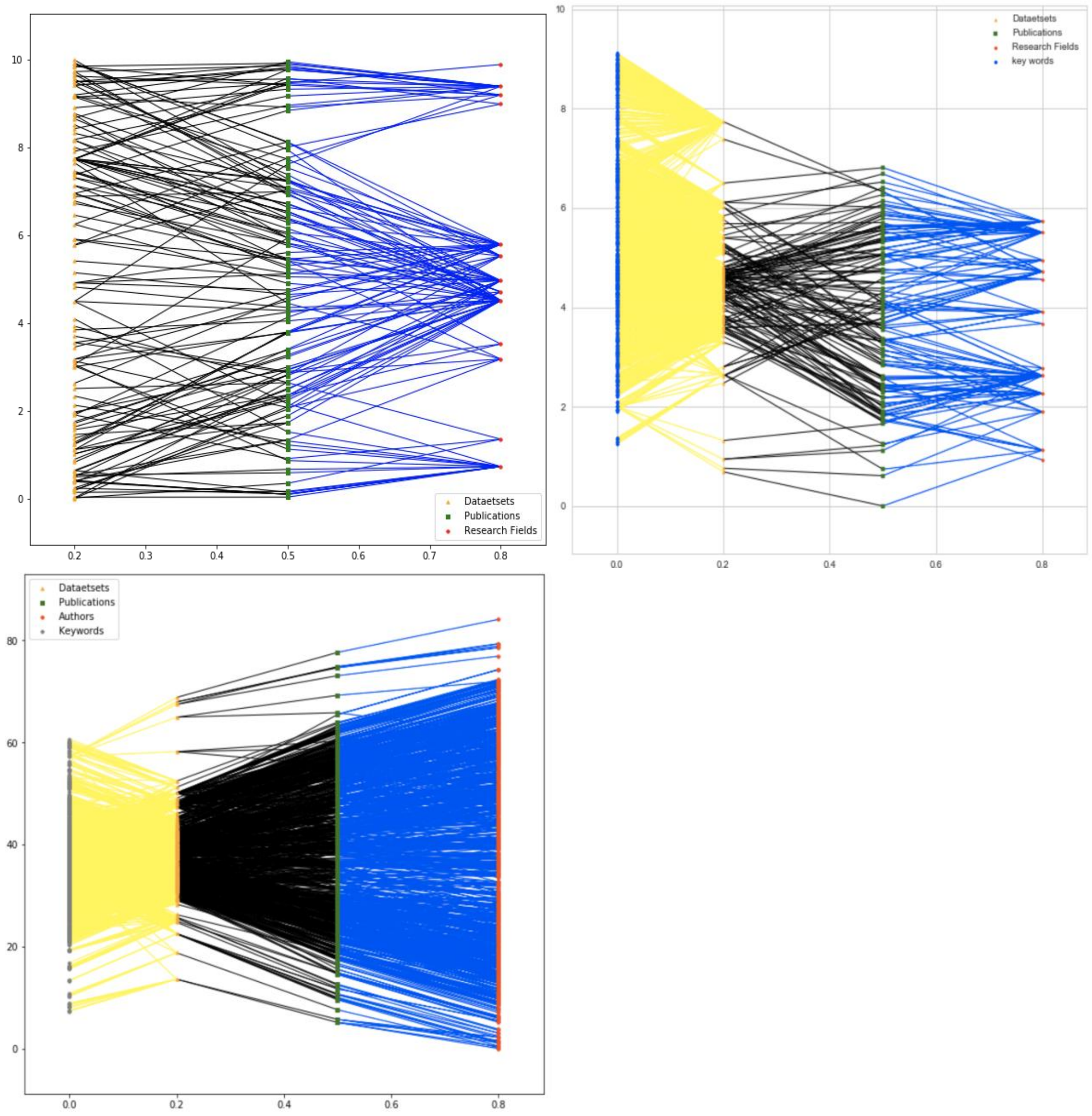


Figure 3. Comparison of network graphs

The graph on the top left displays our first network graph between dataset, publication and research field nodes. The graph on the top right displays our network with additional layer of keywords (subject terms), and the graph at the bottom displays our network between datasets, publications, authors, and keywords(subject terms)

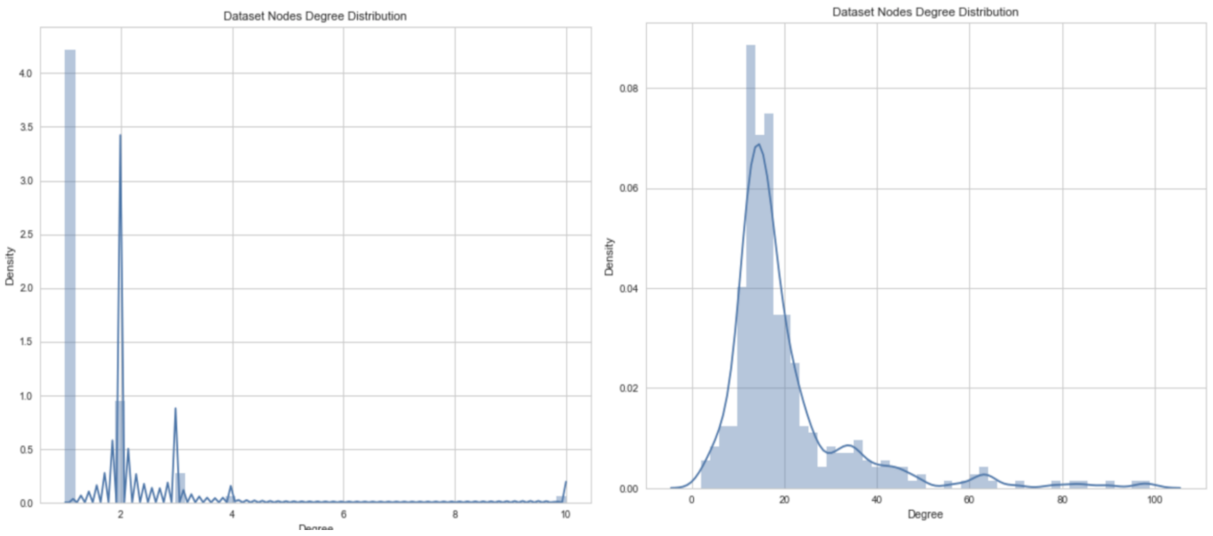


Figure 4. Dataset Nodes degree distribution

The datasets nodes degree distribution of our prototype network is displayed on the left, and the datasets nodes degree distribution of our further developed network is displayed on the right.

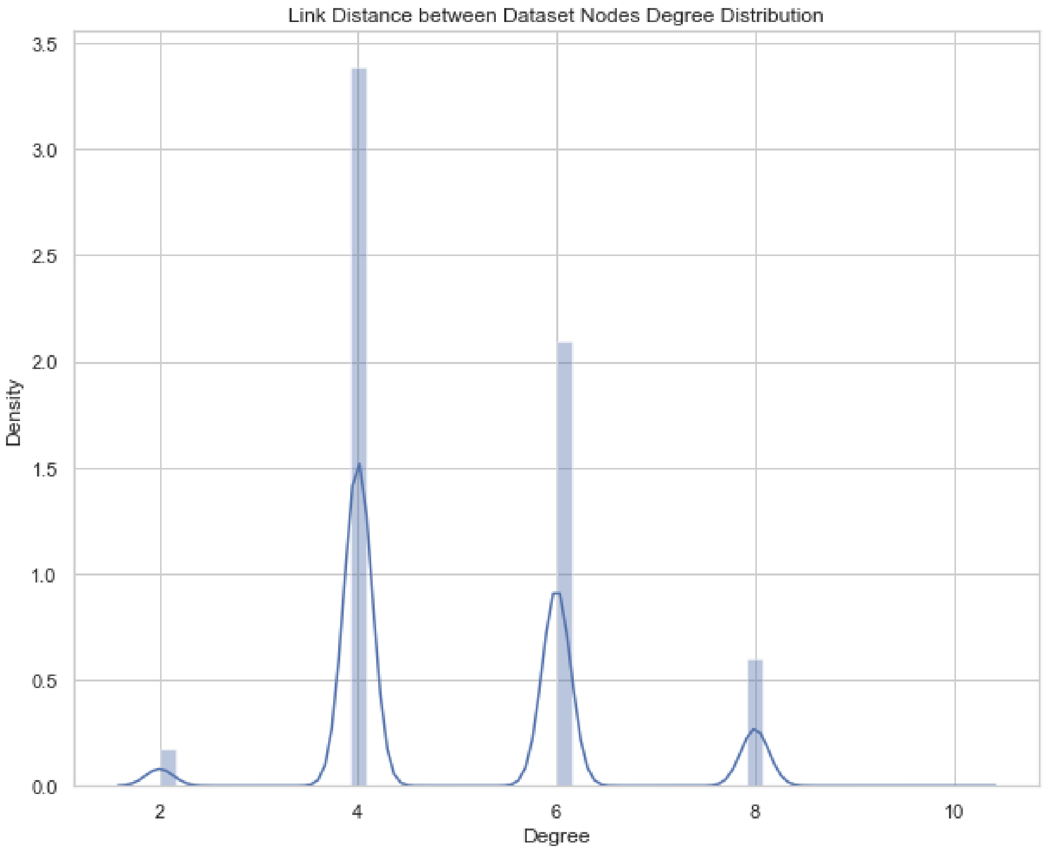


Figure 5. Dataset nodes pairwise shortest distance distribution

These were the only five values of dataset distance we generated using our prototype distance measurement which was not diverse or deep enough for connecting dataset nodes.

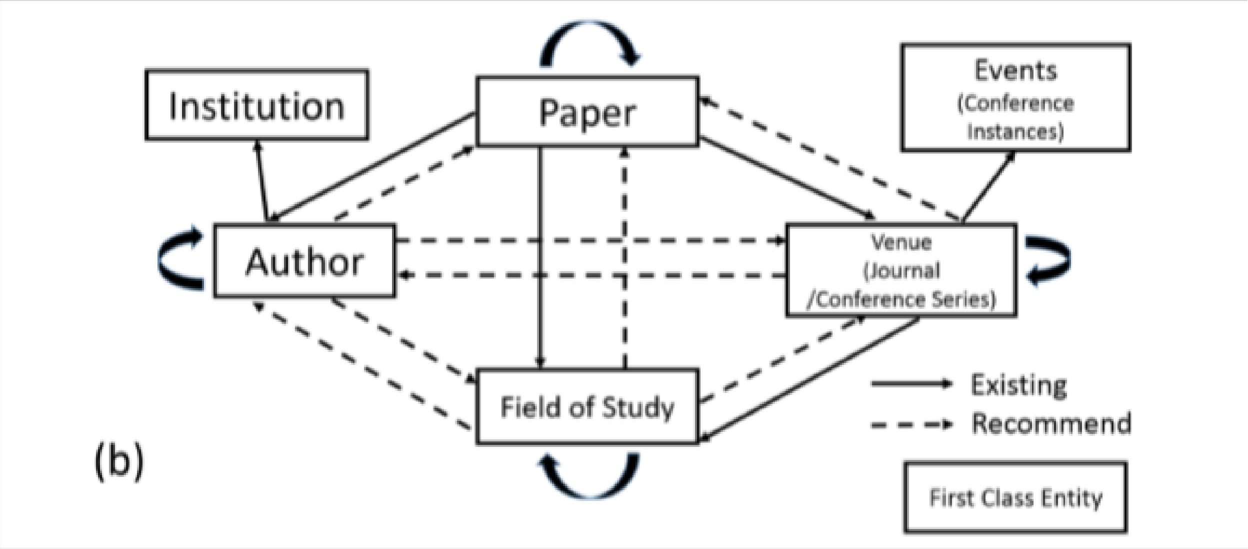


Figure 6. MAG Academic Entity Recommendation Graph (Sinha et al., 2015)
The graph is comprised of six types of entities that model scholarly activities: *field of study, author, institution (affiliation of author), paper, venue (journal and conference series), and event (conference instances)*. Recommendations for homogeneous types of entities can be made from paper to paper, author to author, venue to venue, and field of study to field of study. Recommendations for heterogeneous types of entities can also go from author to paper, venue, and field of study, from field of study to paper, author, and venue, and vice versa.

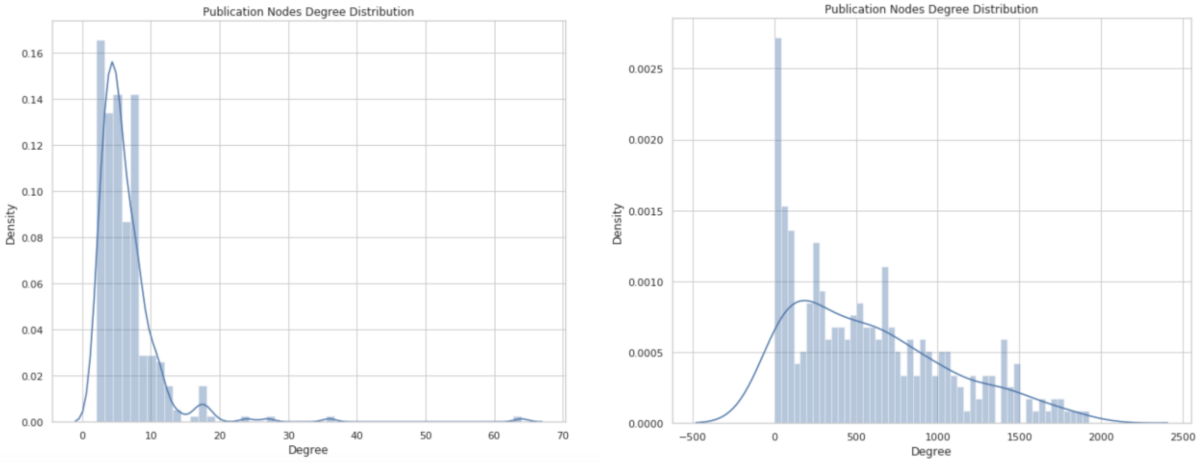


Figure 7. Publication Layer Degree Distribution
Typical degree of publication nodes increases significantly from around 5 to around 100.

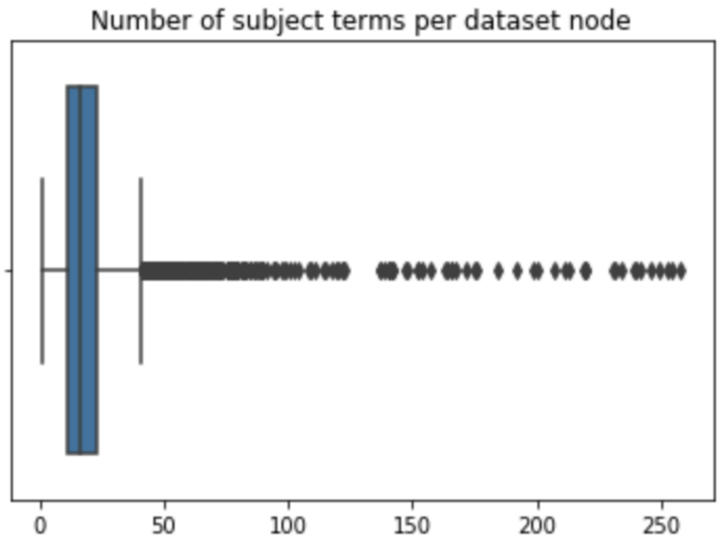


Figure 8. Subject Terms (Keywords) per Dataset Node Distribution
Average number of subject terms per dataset node is 19.

Algorithm	Coverage (%)	Isolated Nodes	Runtime (ms)
Jaccard	37.5	11	381
Cosine	39.6	12	100
Hopcroft	38.2	0	353
Adamic-Adar	3.15	0	755

Figure 9. Link Prediction Algorithm Comparison
Cosine similarity algorithm has the best performance in terms of percent coverage of datasets and runtime for 1000 random simulations. Hopcroft similarity algorithm also has good performance with the second highest percent coverage and 0 isolated dataset nodes.