Geodesic and Neural Features for Link Prediction in the COVID-19 Biomedical Knowledge Graph

Lucas Hurley McCabe
The George Washington University

August 16, 2021

Abstract

Motivated by Challenge 2 of the *5th Annual Smoky Mountains Computational Sciences Data Challenge*, we analyze the COVID-19 biomedical knowledge graph [11]. After computing geodesic statistics for all nodes in the network, we present several machine learning pipelines for automated hypothesis generation. Our most performant approach achieves classification results comparable to the state-of-the-art on thematically similar link prediction benchmarks. Relevant source code and data is available in our public GitHub repository, described in Section 5.

1 Introduction

The volume of new research publications is prohibitively vast, motivating scientific workflows driven by literature-based discovery. In particular, hypothesis generation can reduce research risk by filtering low-probability hypotheses prior to experimentation and hasten the rate of scientific discovery by automating a costly component of the experimental design process [22].

We center our analysis on the COVID-19-related bibliometric knowledge graph generated by Oak Ridge National Laboratory, which represents relationships between biomedical concepts extracted from 435681 publications [13]. For literature-based discovery with knowledge graphs, link prediction using geodesic features can be powerful, but the recent advent of graph machine learning has provided impressive results, as well [15, 6]. In this work, we consider both. Our contributions are severalfold, including both requirements specified by the scientific data challenge and supplemental analysis:

1. Computation and analysis of geodesic statistics, including all-pairs shortest paths (APSP) lengths (Section 2),

2. Machine learning pipelines incorporating geodesic and neural features for biomedical concept link prediction (Section 3),

3. Itemization of our proposed novel connections (Section 4), and

4. Statistical examination of the COVID-19 citation network's adjacency structure (Section 7).

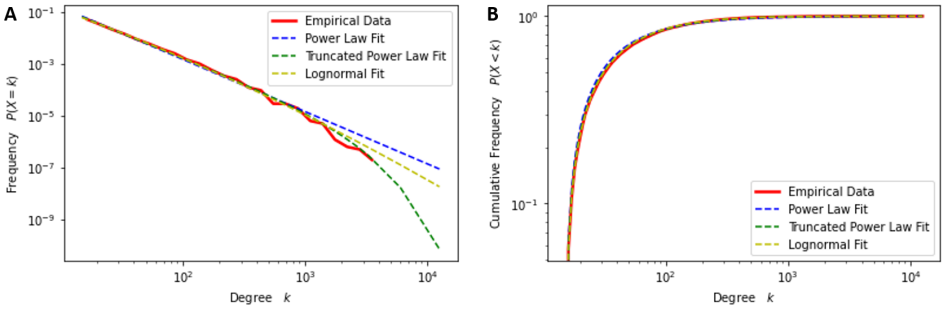

Figure 1: **A)** Log-log plot of G_{tc}'s degree distribution (red) alongside relevant best-fit distributions. **B)** Log-log plot of the undirected and directed (respectively) citation networks' cumulative degree distributions (red) alongside relevant best-fit distributions.

2 Geodesic Statistics

The training data consists of all concept-concept edge relations included in publications up to June 2020, though these are not explicitly specified. Instead, we regard a concept-concept edge relation as *consistent* with a publication if both concepts are neighbors of a publication in the publication-concept network. Since corresponding publication dates are available, we generate the *Training Concept Graph* (G_{tc}) from all concept-concept edge relations whose latest consistent publication was prior to June 2020; this network consists of 46669 nodes, 300673 edges, and has a degree distribution consistent with a truncated power law (Figure 1). We will refer to edges in G_{tc} as *positive training pairs*. The remaining concept-concept edges are withheld as positive samples for validation (we will refer to these 167172 edges as *positive validation pairs*), though we exclude those edges containing

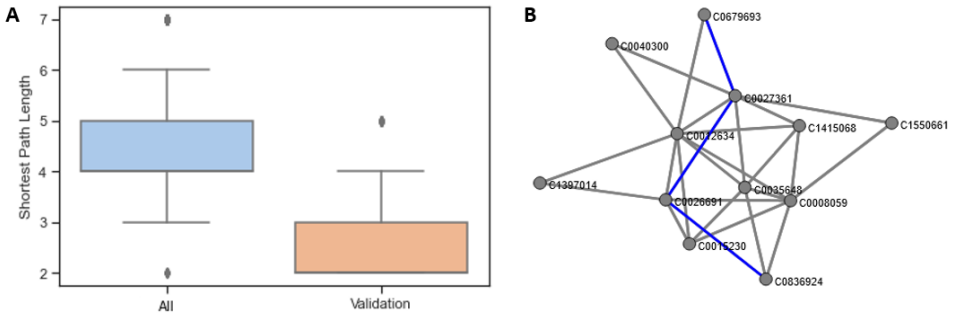

Figure 2: **A)** Boxplot depicting the distributions of shortest path distances between all pairs of nodes in G_{tc} (All) and positive validation pairs in G_{tc} (Validation). **B)** One shortest path (blue) between concepts C0679693 and C0836924 in G_{tc}, illustrated on the induced subgraph of the G_{tc} containing the nodes of the shortest path and randomly-selected neighbors of each node.

nodes not found in G_{tc}, as these correspond to novel conceptual discoveries that cannot be hypothesized in this framework.

2.1 Path Lengths

We compute the APSP lengths for G_{tc} (provided in our GitHub repository, Section 5). While the Floyd–Warshall algorithm is, in principle, the most computationally performant option due to its $\mathcal{O}(|V(G_{tc})|^3)$ time complexity, memory limitations make it intractable for a graph of this size [8, 23, 12]. Instead, we conduct unweighted breadth-first searches for single-source shortest paths on a per-node basis. The average shortest path length among all node pairs in G_{tc} is 4.31 ($\sigma = 0.748$).

We are also interested in the distances between pairs of nodes that will form connections in the future. Toward this end, we measure the shortest path lengths in G_{tc} of positive validation pairs; these correspond to concepts that have recently-established connections not consistent with publications prior to June 2020. The average shortest path length of positive validation pairs in G_{tc} is 2.36 ($\sigma = 0.541$). In general, pairs of nodes that formed connections in the validation period were already near one another.

2.2 Betweenness Centrality

Exhaustive computation of betweenness centrality for large graphs is expensive in terms of both time and space. Instead, we estimate the betweenness

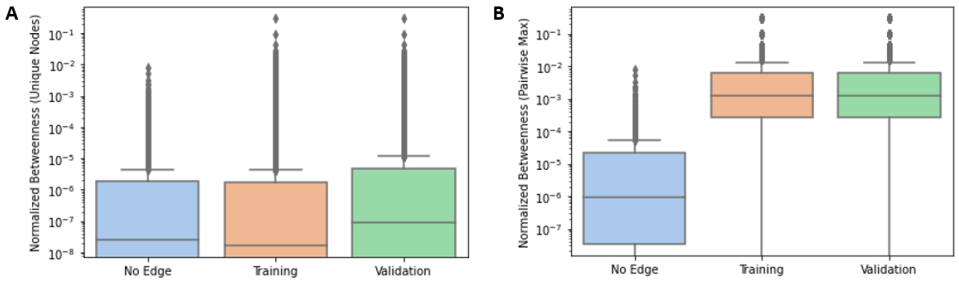

Figure 3: **A)** Log-scale boxplot depicting the distribution of normalized betweenness centrality of all unique nodes found in node pairs who have no edge in training or validation (No Edge), positive training pairs (Training), and positive validation pairs (Validation). **B)** Log-scale boxplot depicting the distribution of pairwise mean normalized betweenness centrality found in node pairs who have no edge in training or validation (No Edge), positive training pairs (Training), and positive validation pairs (Validation).

centrality of each node in G_{tc} using the method of Brandes and Pich with 10000 pivot nodes [3]. Since betweenness scales with node pair count, we normalize all measurements by dividing by the number of node pairs excluding a given node ($(N-1)(N-2)/2$, where $N = |V(G_{tc})|$) [7].

Pairwise mean normalized betweenness centrality tends to be orders of magnitude larger in positive training and validation pairs than in pairs who have no edge in training or validation, but this effect disappears when considering unique nodes only (Figure 3). This is because edges are likely to include high-betweenness nodes, but high-betweenness nodes are not necessarily likely to form edges between one another. As such, betweenness centrality may be a powerful feature for identifying negative examples (hypothesized edges without a high-betweeness node are unlikely), but additional information is necessary for high-precision link prediction.

3 Link Prediction

We implement and assess three link prediction pipelines for automated hypothesis generation:

1. DeepWalk node embeddings [1] are computed based on G_{tc} [20]. We

[1] We replicate the parameters used in the original DeepWalk publication - namely, a dimensionality of 128 with 80 random walks of length 10 [20].

construct two embedding-based pipelines:

(a) *DeepWalk+MLP*: For each hypothesized edge, a multi-layer perceptron [2] estimates the probability of the edge based on the embeddings of each node (Figure 4B).

(b) *DeepWalk+LR*: For each hypothesized edge, we use the Hadamard operator [3] to generate a combined representation of the edge's nodes' feature vectors. We estimate the probability of the edge using this combined representation via logistic regression (Figure 4C).

2. *Geodesic+GBC*: All-pairs shortest paths distances and betweenness centrality metrics are calculated based on G_{tc}. For each hypothesized edge, a gradient boosting classifier [4] estimates the probability of the edge based on the betweenness centrality of each node and the length of the shortest path between them (Figure 4A).

While shortest path length is an informationally-dense feature for link prediction (Figure 2), it provides several practical challenges. First, since the generation of training data warrants sampling positive edges and removing them from the graph, only a small number of positive samples can be used without excessively sparsifying the source data. For this reason, we limit the positive training data for both pipelines to 34668 positive edge samples (corresponding to 10% of Training Concept Graph edges).

Such a procedure may make the graph disconnected, resulting in incalculable shortest path lengths. Where edge removal results in disconnected vertices in the training data, we replace the shortest path length with the effective diameter, defined as the minimum path length whereby 90% of connected nodes can reach one another [16].

4 Results and Discussion

Known concept-concept links are non-exhaustive, unlikely to represent all possible scientific relationships, so our negative examples in training and validation are samples from the edge set of the complement of the graph of existing connections. For this reason, retrieval of true positive links may

[2] hidden layer sizes (64, 12), rectified linear unit activation

[3] Many binary operators are feasible here. We choose Hadamard for its exemplary empirical performance in link prediction tasks with DeepWalk [9].

[4] 100 boosting stages, maximum tree depth 3

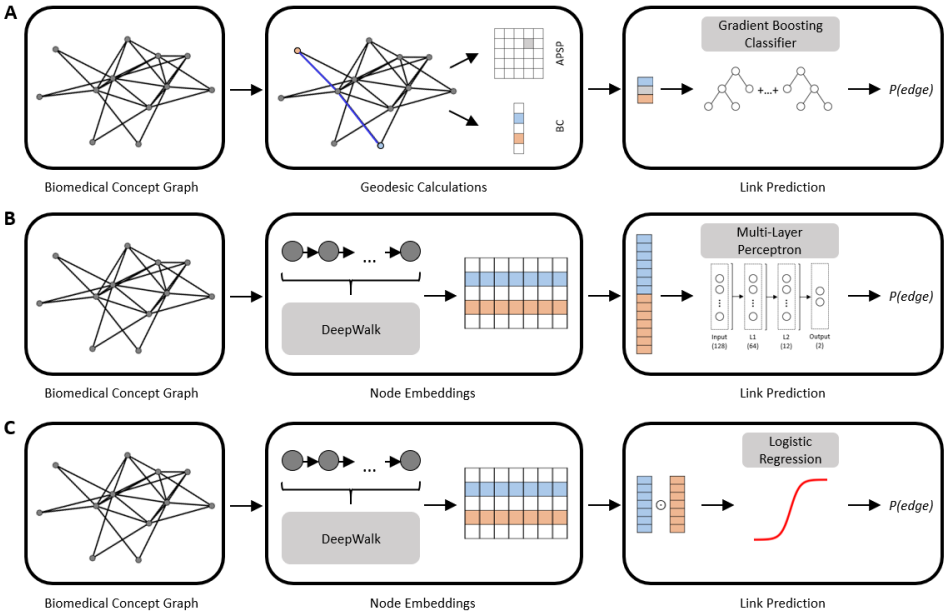

Figure 4: Illustration of our link prediction pipelines: *Geodesic+GBC* (**A**), *DeepWalk+MLP* (**B**), and *DeepWalk+LR* (**C**).

Pipeline	**ACC**	**AUC**	**AP**
DeepWalk+MLP	70.0	77.6	76.4
DeepWalk+LR	74.5	82.1	83.7
Geodesic+GBC	96.8	99.4	99.5

Table 1: Results for the link prediction task on validation data, namely: classification accuracy (ACC), area under ROC (AUC), and average precision (AP).

be more practically important than discrimination of true negative links. Additionally, by this approach, a minor class imbalance problem must be overcome: for all but the densest graphs, the number of existent edges is a small fraction of possible ones.

We evaluate performance based on classification accuracy, area under a receiver operating characteristic curve, and average precision score, as outlined in Table 1. Since this COVID-19 biomedical knowledge graph is a novel dataset without an established state-of-the-art for link prediction, we consider the thematically similar *PubMed* benchmark, for which Pan et. al achieve link prediction results similar to those of our best-performing *Geodesic+GBC* [18]. We also predict novel relations by using *Geodesic+GBC* to rank the highest positive class probabilities of negative samples in validation. The top five predicted novel relations by this method are ('C0035236', 'C1441604'), ('C0027362', 'C0020967'), ('C0003062', 'C0012754'), ('C0086418', 'C0027934'), and ('C0006104', 'C0333230'). Our top 1000 predictions are itemized in the GitHub repository (Section 5).

Geodesic statistics are exceedingly effective features for link prediction, but their advantage comes at significant computational and memory costs for large graphs. For instance, our pre-computed APSP data alone is over 6 GB in size, and such resources may not be available in all use cases. Additionally, as the knowledge graph grows over time, re-computation of features must be repeated from scratch, making the time-consuming task of computing geodesic features entirely unscalable. As such, the more moderately performant *DeepWalk+LR* pipeline may still be better-suited for a practical production environment. For the purpose of guiding scientific experimentation, however, these computations are likely to be infrequent, making the less scalable but higher-precision *Geodesic+GBC* pipeline preferable.

Unresolved in this work is the possibility that the edge-generation procedure may vary over time, especially as scientific research in a particular field

becomes more saturated. For instance, it is conceivable that early connections tend to include the most popular concepts via a procedure like preferential attachment [2], whereas later link formation may be more dependent on path distance or other geodesic statistics. Analysis of this possibility warrants reconstruction of a plausible time series of the COVID-19 biomedical knowledge graph's evolution and is left for future work.

5 Source Code

Source code and processed data are available in our public GitHub repository: `https://github.com/lucasmccabe/smcdc-2021-2`. We also provide *kg_browser*, a convenient utility for accessing our processed data and models.

6 Acknowledgements

Support for DOI 10.13139/OLCF/1782714 dataset is provided by the U.S. Department of Energy, Project smcdc21_challenge2 under Contract DE-AC05-00OR22725. Project smcdc21_challenge2 used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

All graphs are generated using *NetworkX* [10]. Statistical analyses of degree distributions in Sections 2 and 7 are completed using the *powerlaw* library [1]. In Section 3, our classifiers use *Scikit-learn* [19] and our DeepWalk implementation uses *Karate Club* [21].

References

[1] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. powerlaw: a python package for analysis of heavy-tailed distributions. *PloS One*, 9(1):e85777, 2014.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] Ulrik Brandes and Christian Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, 17(07):2303–2318, 2007.

[4] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1–10, 2019.

[5] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[6] Gamal Crichton, Simon Baker, Yufan Guo, and Anna Korhonen. Neural networks for open and closed literature-based discovery. *Plos One*, 15(5):e0232891, 2020.

[7] Donglei Du. Social network analysis: Centrality measures. `https://www2.unb.ca/~ddu/6634/Lecture_notes/Lecture_4_centrality_measure.pdf`. [online lecture notes].

[8] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[10] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[11] Drahomira Herrmannova, Ramakrishnan Kannan, Seung-Hwan Lim, and Thomas E Potok. Covid-19 knowledge graph–dataset for smcdc 2021 challenge 2. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States)., 2021.

[12] Peter Zilahy Ingerman. Algorithm 141: path matrix. *Communications of the ACM*, 5(11):556, 1962.

[13] Ramakrishnan Kannan, Piyush Sao, Hao Lu, Drahomira Herrmannova, Vijay Thakkar, Robert Patton, Richard Vuduc, and Thomas Potok. Scalable knowledge graph analytics at 136 petaflop/s. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–13. IEEE, 2020.

[14] Jinseok Kim. Scale-free collaboration networks: An author name disambiguation perspective. *Journal of the Association for Information Science and Technology*, 70(7):685–700, 2019.

[15] Yong Hwan Kim, Seung Han Beak, Andreas Charidimou, and Min Song. Discovering new genes in the pathways of common sporadic neurodegenerative diseases: a bioinformatics approach. *Journal of Alzheimer's Disease*, 51(1):293–312, 2016.

[16] Jérôme Kunegis. *Handbook of Network Analysis*. 2021.

[17] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.

[18] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. *arXiv preprint arXiv:1802.04407*, 2018.

[19] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.

[21] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, page 3125–3132. ACM, 2020.

[22] Menasha Thilakaratne, Katrina Falkner, and Thushari Atapattu. A systematic review on literature-based discovery: General overview, methodology, & statistical analysis. *ACM Computing Surveys (CSUR)*, 52(6):1–34, 2019.

[23] Stephen Warshall. A theorem on boolean matrices. *Journal of the ACM (JACM)*, 9(1):11–12, 1962.

7 The Network Structure of COVID-19 Citations

Scientific collaboration networks of various kinds have been reported to be scale-free [14]. Here, we generate citation networks from the provided list of paper citations and fit reference distributions to their degree sequences using the methods of Clauset and colleagues [5]. In general, the evidence is consistent with weak power law or log-normal behavior, consistent with Broido and Clauset's observation that "for most networks, log-normal distributions fit the data as well or better than power laws" [4].

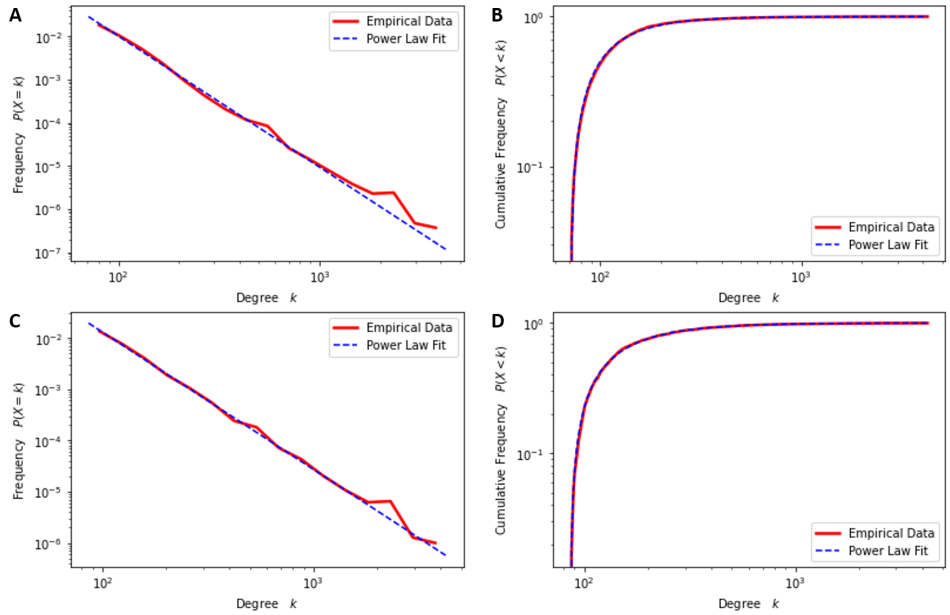

Figure 5: **A, C)** Log-log plot of the undirected and directed (respectively) citation networks' degree distributions (red) alongside best-fit power law distributions (blue). **B, D)** Log-log plot of the undirected and directed (respectively) citation networks' cumulative degree distributions (red) alongside best-fit power law distributions (blue).

7.1 The Undirected Case

In the undirected case, we do not consider the the direction of citations, assessing the general connectivity of the network. We compare the maximum likelihood power law fit ($\alpha = 3.04$) to that of lognormal, exponential, and Weibull distributions, reporting the loglikelihood ratios and relevant p-

values between the power law and alternative distributions in Table 2. We find moderate evidence supporting a power law degree distribution to the undirected version of the citation network (Figure 5 A-B), though we cannot reject a lognormal fit at the 0.05 significance level.

Reference Distribution	**R**	**p**
Log-Normal	0.6462	0.5181
Exponential	7.146	8.910e-13
Weibull	2.840	0.0045

Table 2: Assessment of distribution fit for the undirected citation network, showing log-likelihood ratio (R) of maximum likelihood power law vs. reference distribution alongside corresponding p-values (p).

7.2 The Directed Case

In the directed case, we consider the the direction of citations, analyzing the in-degree distribution of the network. We find a power law exponent of 2.68, similar to the corresponding value found by Newman in biomedical collaboration networks [17]. The in-degree distribution is plausibly power law or log-normally distributed (Figure 5 C-D).

Reference Distribution	**R**	**p**
Log-Normal	-0.0582	0.9536
Exponential	5.938	2.873e-13
Weibull	1.431	0.1526

Table 3: Assessment of distribution fit for the directed citation network, showing log-likelihood ratio (R) of maximum likelihood power law vs. reference distribution alongside corresponding p-values (p).