Finding Foundations

Jack Beasley, jbeasley@stanford.edu Kristine Guo, kguo98@stanford.edu

October 18th, 2018

Introduction

The fields of scientometrics and bibliometrics seek to understand and measure science through quantitative measures of scientific output. Within these fields, determining which papers are most important and impactful within a given field of science is a key problem that requires broad domain knowledge and exhaustive research.

As an alternative to manually seeking out foundational papers, researchers have turned towards citation network analysis in order to examine the evolutionary structure of scientific knowledge throughout time. However, while literature exists on applying these methods to understanding the development of specific fields of science, such findings are limited to specific fields and fail to apply to the broader day-to-day inquiries a researcher has.

To bridge this gap between research and practice, we seek to create a recommendation system based on main path analysis to find articles foundational to the development of the paper at hand, apply those methods to a large, comprehensive citation dataset that draws from many fields, and do so in an efficient manner such that recommendations can feasibly be returned to a user interactively.

By recommender system, we mean a system that takes an arbitrary paper P as an input and returns a set of the k most foundational papers to P in a reasonable amount of time so researchers can in theory use the system interactively. To be useful in the real world, this system must operate on papers from any field, not simply a small subfield, as a subgraph will miss important interdisciplinary citations and only be useful to a small subset of the academic population.

By foundational, we mean papers that contribute the most important ideas to the article in question. A foundational paper is one that contributes ideas that are *used* to create the new article in question, and are often found in lists of "papers that stood the test of time." Reframed, foundational papers are often called classics that lead to the creation of whole fields built on top of their results and would have lead to the greatest negative implications for the current paper if retracted or falsified.

We postulate that these foundational papers can offer researchers and academics information that is different yet complementary to that provided by the articles directly cited by a given paper. For example, having knowledge of the development of the current paper may prove useful when conducting literature reviews, or more generally, when attempting to gain a broad understanding the development of a concept or technique.

Related Work

Single-Score Methods

There exist many different methods to determine node importance in a graph. For paper and journal importance applications, Clarivate Analytics has published journal importance statistics based on two distinct measures: the Journal Impact Factor, which is mostly based on citation counts[1] and the Eigenfactor score,[2] which is essentially a modified PageRank algorithm for citation networks rather than web networks.

While these methods provide effective rankings given the properties they attempt to rank by, these systems do not provide value to the researcher in her day to day work of reviewing literature, as they provide nothing but a single score number that can be hard to interpret. Because these scores distill complex graph phenomena into a single number, they offer little help in determining the actual relationships between articles and understanding the larger development of science. Thus, single-score methods do not effectively help researchers with the problem of determining what the foundations of a paper are.

Because of this lack of information from single-score methods, we seek algorithms that preserve the citation network graph structure. Thus, our research pointed

[1] "Impact Factor - Clarivate."

[2] Bergstrom, West, and Wiseman, "The Eigenfactor™ Metrics."

us towards the usage of main path analysis, which we will explore more in the next three papers.

Combining mapping and citation network analysis for a better understanding of the scientific development[3]

Calero-Medina and Noyons combine bibliometric mapping and citation network analysis in order to investigate the development of scientific knowledge about Absorptive Capacity, a term coined in 1988 that has had widespread influence on the field of Organization.

For citation network analysis in particular, they utilize two different methods: 1) main path analysis, and 2) hubs and authorities analysis. Main path analysis identifies the nodes that are most frequently used in "walks" from the most recent citations to the oldest. By computing all such possible paths, we can discover the papers that are more frequently encountered throughout time, pointing towards their centrality in the development of an academic specialization. This technique is combined with information gained from using hubs and authorities analysis, which identifies papers that are both cited by other prominent papers as well as cite important papers themselves.

By combining these different perspectives, Calero-Medina and Noyons successfully identify 15 papers that comprise the main path component of the Absorptive Capacity field. Thus, this paper provides inspiration for using main path analysis to identify foundational papers in combination with hubs and authorities which actually ranks and scores the papers.

An integrated approach for main path analysis: Development of the Hirsch index as an example[4]

Liu and Lu begin by critiquing the technique of main path analysis. The original main path analysis only identifies a single main path, which is not representative of larger scientific networks that often have multiple main paths. Furthermore, the original algorithm greedily constructs the main path by repeatedly selecting the link with the highest search path count (SPC). However, as with many greedy algorithms, this algorithm is not guaranteed to produce the path with the largest cumulative SPC or contain the link with the largest SPC.

Therefore, Liu and Lu propose new variations on main path analysis. For example, global main path analysis aims to find the path with the true overall largest SPC. Another is multiple main path analysis, which identifies multiple local main paths by relaxing the search constraints to reveal more detailed information. Finally, key-route main path analysis guarantees that the link with the highest SPC is included by beginning the search from both ends of the link instead of the source nodes. Importantly, all of these methods can be combined as well.

Thus, the authors next apply an integrated approach that utilizes a combination of main path analysis methods in order to examine the development of the Hirsch index. Ultimately, their results prove that the main path analyses developed by Liu and Lu enhance our capability to capture different types of information about the relationships between scientific articles.

Knowledge diffusion path analysis of data quality literature: A main path analysis[5]

In this article, Xiao et al. integrate local, global, multiple-global, and key-route main path analyses to uncover knowledge diffusion paths of data quality literature. In particular, they demonstrate that each type of main path analysis reveals different yet complementary information about development trends.

For example, local and global main path analysis highlight the papers that have provided major contributions to the field. On the other hand, multiple global and the key-route main path analyses provide more complete pictures of development trends by identifying multiple paths, revealing the divergence-convergence of the citation network as it evolves throughout time.

Finally, and perhaps most importantly, Xiao et al. also provide intuitive graphical representations of main path analyses in order to both convey their nuances and allow the reader to view the interrelationships between papers. This method of presenting results in particular serves as an inspiration for our project.

Motivation for Improvement

As seen from the literature review above, citation network and main path analysis are often limited to characterizing the development structures of specific concepts and subfields such as Absorptive Capacity, the Hirsch Index, and data quality literature.

[3] Calero-Medina and Noyons, "Combining Mapping and Citation Network Analysis for a Better Understanding of the Scientific Development."

[4] Liu and Lu, "An Integrated Approach for Main Path Analysis."

[5] Xiao et al., "Knowledge Diffusion Path Analysis of Data Quality Literature."

This reality proves less than ideal for researchers and academics, who are told to "stand on the shoulders of giants" but are not given any tools that they can use to efficiently peruse and explore the development of their field. For example, conducting a literature review requires the ability to determine what works constitute essential background reading for a given paper, as well as assessing works with large impact when attempting to create new innovational methods.

However, this is not an easy process, as making literature reviews is a time-consuming manual problem that consists of recursively searching through papers' citations to try to understand what actual authorities and ground truths underlie a research problem. Beyond even just academics, people with casual interest in a field should also be able to have easy access to a field's literature without having to manually search for its most foundational papers. Such tasks would benefit from comprehensive knowledge of the development of the techniques and concepts under question.

Data

Dataset

While there are many options for citation networks, for this project we selected the Microsoft Academic Graph (MAG)[6]. We chose this dataset because it is freely available under an open license, and it has also been described as "the most comprehensive publicly available dataset of its kind" in a review article.[7]

We initially chose the 2017 snapshot of the MAG made available by the Open Academic Society, however, the IDs assigned to papers in that dataset do not match those used by the Microsoft Academic API, meaning that looking up paper titles from IDs required a local copy of the entire uncompressed dataset which totaled 300GB.

Thus, we instead contacted Microsoft and got access to a recent snapshot (accurate as of 2018-10-12) of the current MAG. We then downloaded only the PaperReferences file, which is a 31.3 GB edge list, where paper IDs are 64-bit integers.

Efficient Subgraph Construction

The MAG is a fairly large graph and is certainly much larger than can comfortably run on our laptops with 8GB of RAM. Thus, we needed to devise a method of efficiently generating a subgraph by doing a breadth-first search starting from a paper of interest, without putting the whole graph in RAM.

There has been some work in this space tackling the general problem of working with graphs that don't fit in RAM, such as GraphChi and X-Stream,[8] whose approaches are quite robust and comprehensive as they aim to do general graph analysis with data that doesn't fit in RAM. However, for our project we are only concerned with finding and constructing the neighborhood of any given paper, which can fit in RAM.

In other words, we must devise a method for the initial construction of the subgraph by BFS from the paper of interest. To do so, we devised a system that efficiently crawls over the entire MAG edge list and outputs an edge list that corresponds to a specific paper using the principles proposed by GraphChi and X-Stream. Our system has two distinct phases, preprocessing and BFS construction, which we will describe in two sections.

Preprocessing

We adopt a simplified version of the concept of "shards" proposed by GraphChi in order to simplify our implementation modeled after a hash map. The algorithm we implemented reads through every edge in the MAG's edge list and writes edges to files based on a simple modular hash function on either the source paper's ID or the referenced paper's ID. For example, if an edge has a source ID of 23 and a destination ID of 15 and we are hashing into 20 buckets, we would write that entry once to a file called 03.txt for the source-indexed hash map and once to a file called 15.txt for the destination-indexed hash map.

For the MAG, we decIDed to create both a source and a destination hash map with 3000 buckets each. Each of these contains a full copy of the MAG edge list and thus this data structure consumes a total of 62.6 GB of space on the disk, which fits on our laptops. We chose 3000 buckets, because that was the general heuristic for the largest number of file descriptors allowed open at one time. The number of open file descriptors is a limiting factor in this case because the preprocessing program maintains an open file and corresponding buffer for each output bucket to speed up the preprocessing step, which takes around 15m for each hash table. Each of the 3000 buckets are around 10MB.

[6] Sinha et al., "An Overview of Microsoft Academic Service (MAS) and Applications."

[7] Herrmannova and Knoth, "An Analysis of the Microsoft Academic Graph."

[8] Roy, Mihailovic, and Zwaenepoel, "X-Stream," @kyrola-GraphChiLargeScaleGraph.

Traversal

In the traversal step, we wrote a program that starts from a single paper, then recursively finds that papers transitive references, exploiting the fact that the hash map file structure guarantees that all the in or out references for a given paper will be found in the same 10MB file. This means that finding all the references for a node requires reading only 10MB rather than the whole 31.3 GB edge list.

This crawler can be broken up into three distinct functions. The first is finding references within a file and constructing hash map from paper IDs to sets of corresponding referenced paper IDs for each matching edge. The second is efficiently running that reference finding routine for a set of papers, ensuring that each file bucked is visited only once and merging the resulting data structures. The final routine manages the breadth-first search, keeping track of a set of paper IDs at the current level as well as a set of seen paper IDs and merging the results of successive calls to the references of routine for the set of papers at each level of the BFS.

The finding references in a file routine first creates a buffer that is 10MB in size and begins loading the whole bucket of edges into memory, then scans through line by line in the file and generates a map from 64-bit integer paper IDs to sets of 64-bit integer referenced paper IDs. We initially had an issue where the function in Go to parse string paper IDs into 64-bit integers resulted in a heap allocation of the string value of the ID which we fixed by writing our own integer parsing function that was entirely stack-allocated, increasing parsing performance consIDerably due to reduced pressure on Go's garbage collector.

The routine that finds the references of a set of papers first creates a hash map that maps file bucket names to a set of paper IDs to look for in that file. This aggregates the paper IDs to ensure that each file is only read once. The routine then spins up a lightweight thread (Goroutine) for each file and run the reference finding routine detailed above for each file concurrently. After all the references have been found, the results from each of the files are aggregated into a single hash map.

The BFS routine begins by creating a set of visited papers and a set of current papers, each initialized with the seed paper ID. The routine then makes a call to the routine detailed above to find all the paper IDs referenced by the papers in the current set. The references found by that routine are added to the visited papers set and the unseen papers found are added to the current papers set. The reference finding routine is then called on the new current papers set and the process repeats until a user-set level limit is reached. After this process concludes, the resulting hash maps from each level are aggregated into a single representation of the subgraph found by the BFS and that subgraph is output as an edge list to a file.

This method has proven quite successful for our use case. A naive implementation would have needed to traverse the entire edge list for each level of BFS. Using a highly optimized text search program ripgrep,[9] finding the references of a paper (MAG ID 252001) takes 70.17 seconds. Using our crawler on the hashed dataset, the same operation takes 1.41 seconds.

In our final report, we plan to delve a bit more deeply into the performance characteristics of our crawler, consIDering questions such as how often do reference searches reference the same file and how does performance scale to larger subgraphs with varying node and edge counts. This performance data can be collected concurrently with an exploration of the MAG itself in a style similar to Broder et. al.'s analysis of the web.[10] To explore the MAG, we will need to BFS on a lot of papers and while we do that, we will also collect the CPU, disk and memory metrics for each run of our BFS program to try to understand how it scales.

Initial Observations

One challenge we faced was, surprisingly, dealing with reciprocal edges. Although citation networks are theoretically directed acyclic graphs, we found pairs of papers that cited each other. This implies that a paper cited a paper that had not yet been published at the time of publication. We found an instance where a patent was reissued and its citations updated, but the original published date left unmodified. Additionally, we found academic papers that cited papers that had yet to be published, presumably because the papers were done concurrently and one was published slightly before the other.

To deal with these pairs, we fetched the dates of both papers using the Microsoft Academic API and deleted the edge from the more recent paper to the older paper. While this worked for our initial analysis, we found that these situations are much more common than we previously anticipated and thus we plan to find a way to handle these reciprocal edges int our analysis methods so that we don't need to trim references

[9] Gallant, "Ripgrep Is Faster Than {Grep, Ag, Git Grep, Ucg, Pt, Sift} - Andrew Gallant's Blog."

[10] Broder et al., "Graph Structure in the Web."

from the graph using slow API calls. In general, our methods need to take into account citation cycles, either through preprocessing or adapting our methods. This will be a major focus for our analysis for our final report.

Methods

We will evaluate the various methods for determining paper importance in a citation network, including popular single-score centrality-based methods, common heuristics, and main path analysis. Then, we will compare their performance in order to determine which performs the best for finding foundational papers.

Single-Score Importance

As a baseline, we will investigate the performance of popular single-score methods of measuring importance. Specifically, we will survey node degree (number of citations) as a common heuristic for determining a paper's importance, as well as two measures of node centrality: PageRank and Hubs and Authorities.

Node Degree

Intuitively, important papers are cited more often than the average paper, and thus one heuristic we can use to find foundational papers is simply recommend the nodes with the greatest in-degree links, i.e., the nodes with the greatest number of citations.

PageRank

The PageRank for any paper

Hubs and Authorities

In application to citation networks, Hubs and Authorities assigns each paper two scores: a hub score, which measures how many authoritative papers the given work references, and an authority score, which measures the importance of the paper's contributions and information.

Main Path Analysis

There are two major algorithmic decisions to make when implementing main path analysis: choosing a method of finding traversal counts and choosing a path search mechanism. Notably, we needed to implement from scratch all the main path analysis methods below.

Traversal Counts

There are many different methods for computing edge traversal counts, but most yield similar results in practice. However, previous literature suggests that search path count (SPC) provides additional favorable properties on top of otherwise similar methods, and therefore was the most preferred and widely used.[11] For this reason, we plan to adopt SPC.

The SPC value for a given edge is the number of times it is traversed during all possible paths from source to destination nodes. Manually computing all possible paths in a graph is computationally expensive, but fortunately Batagelj[12] devised an efficient algorithm to compute the SPC values of all edges in $O(\#\text{ of edges})$ time.

Let aRb represents an edge from node a to b. We define two new quantities:

$$N^-(u) = \begin{cases} 1, & u = s \\ \sum_{v:vRu} N^-(v), & \text{otherwise} \end{cases}$$

Where $N^-(u)$ denotes the number of paths from the source node s to node u.

$$N^+(u) = \begin{cases} 1, & u = t \\ \sum_{v:vRu} N^+(v), & \text{otherwise} \end{cases}$$

Where $N^+(u)$ denotes the number of paths from v to a destination node t (nodes with no out-links).

Thus, we topologically iterate over all the nodes and compute these two values. Then, the SPC value of edge (u, v) can be computed by $N(u, v) = N^-(u)N^+(v)$.

Path Search

Unlike for traversal counts, different path search methods can yield significantly different results. There are three common techniques: local search, global search and key-route search. We plan to evaluate these different approaches on our dataset and select the one that seems to work the best for our specific case of identifying foundational papers.[13]

So far, we have implemented local search for main path analysis. Starting from the source node (i.e.,

[11] Xiao et al., "Knowledge Diffusion Path Analysis of Data Quality Literature."

[12] Batagelj, "Efficient Algorithms for Citation Network Analysis."

[13] Xiao et al., "Knowledge Diffusion Path Analysis of Data Quality Literature."

the paper under consideration), this search greedily follows the edge with the greatest SPC value until it reaches a destination node. The paper that the search encountered during its traversal make up the main path, which are then reported as the foundational papers.

Finally, with the list of unique paper IDs identified by main path analysis, we utilize API requests to Microsoft Academic to retrieve the papers' titles and years. Thus, the final output of the algorithm is a list of papers that are easily human readable and interpretable.

Initial Results

We found the subgraph around Grover and Leskovec's paper "node2vec: Scalable Feature Learning for Networks"[14] using a BFS following only in-links (citations) for 2 steps. The resulting subgraph had 1395 nodes and 1684 edges. The table below presents the results from each of the methods detailed above (all results besides MPA ranked in descending order):

Results

All papers recommended based on "node2vec scalable feature learning for networks"[15] and all titles copied verbatim from the MAG titles.

Main Path Analysis

1. "grarep learning graph representations with global structural information" (2015)
2. "line large scale information network embedding" (2015)
3. "deepwalk online learning of social representations" (2014)
4. "representation learning a review and new perspectives" (2013)
5. "a global geometric framework for nonlinear dimensionality reduction" (2000)
6. "image representations for visual learning" (1996)

Most Citations

1. "friends and neighbors on the web" (2003)
2. "nonlinear dimensionality reduction by locally linear embedding" (2000)
3. "the link prediction problem for social networks" (2007)
4. "reducing the dimensionality of data with neural networks" (2006)
5. "a global geometric framework for nonlinear dimensionality reduction" (2000)
6. "a neural probabilistic language model" (2003)

PageRank

1. "nonlinear dimensionality reduction by locally linear embedding" (2000)
2. "a global geometric framework for nonlinear dimensionality reduction" (2000)
3. "normalized cuts and image segmentation" (2000)
4. "spectral graph theory" (1996)
5. "higher eigenvalues and isoperimetric inequalities on riemannian manifolds and graphs" (2000)
6. "the link prediction problem for social networks" (2007)

Authorities

1. "collective dynamics of small world networks" (1998)
2. "the strength of weak ties" (1973)
3. "hierarchical structure and the prediction of missing links in networks" (2008)
4. "finding community structure in very large networks" (2004)
5. "maps of random walks on complex networks reveal community structure" (2008)
6. "self organization and identification of web communities" (2002)

Hubs

1. "community detection in graphs" (2010)
2. "overlapping communities explain core periphery organization of networks" (2014)
3. "leveraging social media networks for classification" (2011)
4. "supervised random walks predicting and recommending links in social networks" (2011)
5. "the link prediction problem for social networks" (2007)
6. "rolx structural role extraction mining in large graphs" (2012)

Discussion

While we are still working on defining a set evaluation method, the differences between the results produced by main path analysis and the other methods is apparent. For example, the main path analysis results include the paper "DeepWalk: Online learning of social representations" by Perozzi, Al-Rfou, and Skiena (2014), which is cited in the original node2vec paper

[14] Grover and Leskovec, "Node2Vec."
[15] Grover and Leskovec.

as an important but inflexible baseline upon which to improve feature representation.

Furthermore, main path analysis allows for the added dimension of being able to track the development of papers and visualize the scientific flow of knowledge that culminates in our original paper. Thus, while other results have papers that are popular yet only adjacently related to both each other and the original paper, the main path analysis results offer a more focused view on the literature that is more pertinent to the particular paper under examination than to others.

Further Work

At this point in our project, we feel confident in our workflow for working with the MAG and with our general implementation of SPA and main path analysis. Specifically, our work on processing the data and constructing paper's local subgraphs through BFS has allowed us to run our algorithms on a vast range of inputs. Furthermore, we have laid the groundwork for further baseline improvements by implementing our own version of main path analysis, which we can now iterate on and improve.

In the remainder of our project we plan to shift our focus broadly from implementation to evaluation and refinements of our current methods. We will quickly outline our plan for the remainder of the project in each area of the project.

For data, we plan on running a survey of the MAG to get an idea for the distributions of subgraphs that stem from papers. This will involve finding the distributions of stats like clustering coefficients, number of nodes and number of edges of these subgraphs. While doing this survey of the MAG, we also plan to use and evaluate our crawler, collecting runtime, cpu and memory use and other performance metrics for each subgraph we crawl. We expect this survey to be a relatively small project that will give us some better intuitions for the structure of the MAG.

We plan to focus most of our energy for the rest of the project on fine-tuning our recommendations algorithms and evaluating them against several baselines to get an idea of how our recommendations differ from those given by methods like PageRank.

Our first project is to stabilize our implementation of SPC. Right now, some edges have SPC values of 0, which we believe to be the result of cycles in the citation network. This particular phenomenon needs more research into its exact causes.

Once we are confident in our SPC implementation, we plan on implementing several different flavors of path search so see how their results differ. We currently implement local search, but we plan to also evaluate key-route and global search. As a part of this project, we want to find an effective way of visualizing these main paths to help us evaluate their effectiveness. Additionally, for long main paths, we want to find a good heuristic, like PageRank or HITS, to select the most relevant articles on that path.

Once we have solidified our implementation of main path analysis recommendations, we plan on evaluating it against several baselines, including the ones we have already done as well as other, more sophisticated baselines, like node2vec. Evaluation will require labeling several papers manually that we know well, to determine which papers we think are most foundational and therefore good recommendations and determining how many of those each method finds. We acknowledge that this will be somewhat of an art as the question of what is foundational will depend on who is asked, however, we think this method, along with listing all the results, will give us a pretty good idea of how the different systems perform. Additionally, if time permits, we might try to find recommendations using different systems for certain papers and ask the authors of those papers which set of recommendations identified what they think are the most foundational papers. We anticipate that evaluation will be tricky as success for a recommender system is hard to define without a large number of people using said recommender system, so we plan to put a lot of energy into this aspect of our final report.

References

Batagelj, Vladimir. "Efficient Algorithms for Citation Network Analysis," September 13, 2003. http://arxiv.org/abs/cs/0309023.

Bergstrom, Carl T., Jevin D. West, and Marc A. Wiseman. "The Eigenfactor™ Metrics." *Journal of Neuroscience* 28, no. 45 (November 5, 2008): 11433–4. https://doi.org/10.1523/JNEUROSCI.0003-08.2008.

Broder, Andrei, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. "Graph Structure in the Web." *Computer Networks*, 2000, 12.

Calero-Medina, Clara, and Ed C. M. Noyons. "Combining Mapping and Citation Network Analysis for a Better Understanding of the Scientific Development: The Case of the Absorptive Capacity Field." *Journal of Informetrics* 2, no. 4 (October 1, 2008): 272–79. https://doi.org/10.1016/j.joi.2008.09.005.

Gallant, Andrew. "Ripgrep Is Faster Than {Grep, Ag, Git Grep, Ucg, Pt, Sift} - Andrew Gallant's Blog." Accessed November 9, 2018. https://blog.burntsushi.net/ripgrep/.

Grover, Aditya, and Jure Leskovec. "Node2Vec: Scalable Feature Learning for Networks." In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–64. KDD '16. New York, NY, USA: ACM, 2016. https://doi.org/10.1145/2939672.2939754.

Herrmannova, Drahomira, and Petr Knoth. "An Analysis of the Microsoft Academic Graph." *D-Lib Magazine* 22, no. 9/10 (September 2016). https://doi.org/10.1045/september2016-herrmannova.

"Impact Factor - Clarivate." Accessed October 18, 2018. https://clarivate.com/essays/impact-factor/.

Kyrola, Aapo, Guy Blelloch, and Carlos Guestrin. "GraphChi: Large-Scale Graph Computation on Just a PC," n.d., 16.

Liu, John S., and Louis Y. Y. Lu. "An Integrated Approach for Main Path Analysis: Development of the Hirsch Index as an Example." *Journal of the American Society for Information Science and Technology* 63, no. 3 (March 1, 2012): 528–42. https://doi.org/10.1002/asi.21692.

Roy, Amitabha, Ivo Mihailovic, and Willy Zwaenepoel. "X-Stream: Edge-Centric Graph Processing Using Streaming Partitions." In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles - SOSP '13*, 472–88. Farminton, Pennsylvania: ACM Press, 2013. https://doi.org/10.1145/2517349.2522740.

Sinha, Arnab, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. "An Overview of Microsoft Academic Service (MAS) and Applications." In *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*, 243–46. Florence, Italy: ACM Press, 2015. https://doi.org/10.1145/2740908.2742839.

Xiao, Yu, Louis Y. Y. Lu, John S. Liu, and Zhili Zhou. "Knowledge Diffusion Path Analysis of Data Quality Literature: A Main Path Analysis." *Journal of Informetrics* 8, no. 3 (July 1, 2014): 594–605. https://doi.org/10.1016/j.joi.2014.05.001.