2019 IEEE International Conference on Big Data (Big Data)

Paper Recommendation Based on Citation Relation

William Tanner
Department of Mathematics and Computer Science
Austin College
Sherman, Texas
wtanner16@austincollege.edu

Esra Akbas
Department of Computer Science
Oklahoma State University
Stillwater, Oklahoma
eakbas@okstate.edu

Mir Hasan
Department of CSIT
Austin Peay State University
Clarksville, Tennessee
hasanm@apsu.edu

***Abstract*—Searching for relevant literature is a fundamental part of academic research. The search for relevant literature is becoming a more difficult and time-consuming task as millions of articles are published each year. As a solution, recommendation systems for academic papers attempt to help researchers find relevant papers quickly. This paper focuses on graph-based recommendation systems for academic papers using citation networks. This type of paper recommendation system leverages a graph of papers linked by citations to create a list of relevant papers. In this study, we explore recommendation systems for academic papers using citation networks incorporating citation relations. We define citation relation based on the number of times the origin paper cites the reference paper, and use this citation relation to measure the strength of the relation between the papers. We created a weighted network using citation relation as citation weight on edges. We evaluate our proposed method on a real-world publication data set, and conduct an extensive comparison with three state-of-the-art baseline methods. Our results show that citation network-based recommendation systems using citation weights perform better than the current methods.**

***Index Terms*—Citation networks, recommendation systems, scholarly data, paper recommendation, network science**

I. Introduction

The academic community publishes millions of research articles each year, and the rate of publications is increasing over time [1]. With an increasingly large body of work, the literature search is becoming a challenging and time-consuming task. Some common ways researchers find literature are searching for keywords and topics in online scholarly databases, such as, Google Scholar, CiteSeer, etc., looking for references in relevant papers they have already found and read, or looking for references used by experts in the field. It is possible to combine techniques to do a well-rounded search. For example, researchers using Google Scholar can search contents for keywords, see which papers it is cited by, and see related papers. However, all these techniques take time, and cannot be used to examine a large amount of literature efficiently.

In addition to being inefficient, these approaches suffer from other weaknesses as well. Keyword-based searches can fail to account for identical concepts with different names in different fields or find unrelated concepts with the same name in different fields [2]. Since it is impossible for a published paper to cite something that has yet to be written, browsing published papers for references will only lead a researcher backward in time.

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

For these reasons, academic paper recommendation systems emerged as a field of research beginning in 1998 with the project CiteSeer, a paper indexing system [3]. There are several classes of paper-recommendation systems, including content-based filtering (CBF), collaborative filtering (CF), graph-based, and hybrid. Each one attempts to measure the relevance among research papers using different approaches.

Our research focuses on graph-based recommendation systems using a citation network that connects papers. Edges connecting papers are based on citation relations between papers. It is also possible to define different edge types based on author, venue, and text similarity, and create multi-level graphs by using a combination of edge types. Once a graph is constructed, an algorithm will provide recommendations using graph metrics. Common examples include random walk with restart (RWR), bibliographic coupling (BC), and co-citation inverse document frequency (CCIDF).

In a citation network, not all citations are equally relevant. An author cites a variety of sources. Some sources might be related to basic algorithms or standard methods, or they may be survey articles. These are cited in the content infrequently. On the other hand, more relevant sources which focus on a similar research problem might be cited more frequently, as the author might compare their methods, techniques, and results. Citation networks with unweighted edges or with edge-weights based on other factors disregard this information. In order to differentiate citations covering the above cases and measure the strength of relation between papers, we incorporate the relevance of citation relations in our recommendation system. The relevance is defined as a citation weight based on the number of times the origin paper cites the reference paper.

Our contributions in this work include the following:

- We propose a method to capture the strength of the relevance between citing and cited papers. We define a citation weight based on the number of times the citing papers cites the reference papers as the strength of the relevance in a citation graph. Then, we create a weighted citation network based on citation weight.
- We apply different graph based recommendation methods with modifying them to work on the weighted network.
- We conduct an experiment of studies with a real-world publication dataset. From an unweighted citation net-

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

work, we use the paper IDs and titles to get the PDF of each paper. After processing text of PDF, we extract the citation parts from text. We compute citation weights and create a weighted citation network. Our experiments show promising results to find relevant papers.

The rest of the paper is organized as follows: In Section II, we discuss the related work for the paper recommendation system by categorizing them into five different groups. In Section III, we introduce our complete recommendation system. Experimental studies and key findings are reported in Section IV, followed by concluding remarks in Section V.

II. Related Work

There has been significant research in paper recommendation [4]. Different recommendation techniques have been proposed, which can be categorized as follows: (A) content-based approaches, (B) collaborative filtering-based approaches, (C) citation-based approaches, (D) graph-based approaches, and (E) hybrid approaches.

A. Content-based Approaches

Content-based recommendation algorithms compute the content similarity of the user query/the input paper and the papers in the corpus to find relevant papers to recommend. Different textual information, such as title, abstract, bibliography, author provided keywords, ACM classification tree, as well as the paper's whole body text are utilized to compute the similarity score [5] [6] [7] [8] [9] [10]. Content-based filtering techniques depend highly on accessing the contents of the papers. Some research shows that weighting words from the title, abstract, and body of the paper differently can improve accuracy [11] [12] [13]. However, extracting text from PDFs is challenging, and can introduce errors. The challenges we face when processing PDFs, and the techniques to handle those challenges will be further discussed in section III(A). Another issue with content-based comparisons is that it fails to account for identical concepts with different names in different fields or find unrelated concepts with the same name in different fields [2].

B. Collaborative Filtering-based Approaches

Collaborative filtering recommendation algorithms provide recommendations for a user based on similar users' preferences. Some of the existing works in paper recommendations apply different collaborative filtering techniques to find relevant research papers [14] [15] [16]. In paper recommendations, rating of papers can be used as a user's preference measure, but research finds that users are unwilling to explicitly rate research papers [14] [16] [17]. To overcome this, some of the existing works interpret users' interactions with items (e.g., number of pages read, downloading a paper, viewing a bibliography, etc.) as implicit ratings [16] [18] [15]. Another issue with collaborative filtering in the field of research paper recommendation is the ratio of researchers to research papers. Collaborative filtering is ineffective in domains where more items than users exist because the chances of users interacting with the same items is smaller [19].

C. Citation-based Approaches

Citation information can be utilized to compute the relatedness among academic papers [20]. Some of the existing approaches apply different citation analysis techniques, such as co-citation analysis [21], bibliographic coupling [22], or citation proximity analysis [23] to find relevant papers for an input paper or to get citation recommendations for a research topic. Citation databases, such as Google Scholar and CiteSeer utilize citation counts and different citation analysis techniques to identify papers that are similar to an input paper [24].

D. Graph-based Approaches

Graph-based methods build networks that connect papers. Edges connecting papers can be based on citations [20] [25] [26], authors [27] [25], topics or keywords [25], text similarity [28], and so on. It is also possible to create multi-level graphs by using a combination of edge types. Once a graph is constructed, an algorithm provides recommendations using graph metrics. Common examples include random walk with restart (RWR) or utilizing different citation metrics, such as bibliographic coupling and co-citation. There are existing networks, such as Microsoft Academic Graph (MAG) [29], ACL Anthology Network [30], and SNAP Network Dataset [31], which can be utilized to develop a graph-based recommender system.

E. Hybrid Approaches

Hybrid approaches use a combination of techniques to provide recommendations. For example, a CBF recommendation system can use a graph to restrict potential recommendations [3]. Since all recommendation systems have inherent strengths and weaknesses, hybrid approaches combine techniques in an attempt to create well-rounded recommendations.

All citations in a paper are not equally important, some citations are more relevant than the others [32]. For example, one paper might be cited only once in a paper; on the other hand, another paper might be cited multiple times, which indicates that those papers are more related. Papers that are cited together can also be an indication of their relevance. In the proposed method, we utilize an unweighted paper citation network to create a weighted citation network based on citation analysis, and implement several recommendation algorithms on both networks to find relevant papers. We collect the documents listed in the unweighted paper citation network, analyze the paper body to extract the number of times the citing papers cite the reference papers to measure their relevance, and assign that number as their edge weight in the unweighted citation network.

III. Our Approach

In this paper, we propose a recommendation system to find relevant papers for a given input paper which utilizes a *weighted* paper citation network that we created. Figure 1 depicts the system architecture of the proposed method.

Our system includes the following steps:

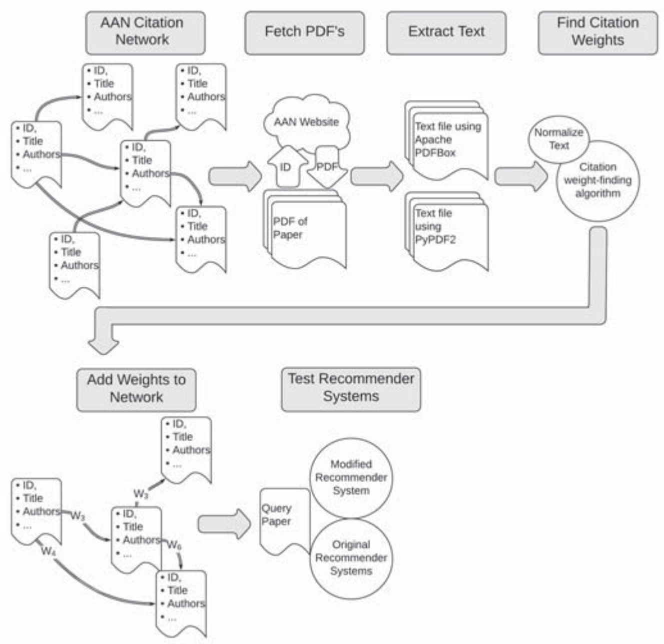
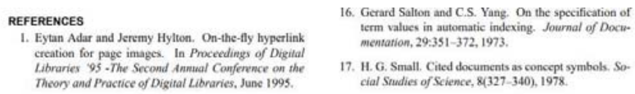


Figure 1: System Architecture of the Proposed Method



REFERENCES

1. Eytan Adar and Jeremy Hylton. On-the-fly hyperlink creation for page images. In *Proceedings of Digital Libraries '95 -The Second Annual Conference on the Theory and Practice of Digital Libraries*, June 1995.

16. Gerard Salton and C.S. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372, 1973.

17. H. G. Small. Cited documents as concept symbols. *Social Studies of Science*, 8(327–340), 1978.

Figure 2: Sample Paper Written in Two-column Format [33]

1) For a given unweighted network, we fetch the PDF of each paper in the network. After collecting all PDF files of papers, we translate the PDFs into text files using PyPDF2 and Apache PDF Box, and normalize the text using different natural language processing techniques, such as uppercase to lowercase conversion, stop word removal, and stemming.
2) We extract citation information from text using a combination of regular expressions, and count the number of times a paper is cited in a paper for each cited paper.
3) We create the weighted network by adding the number of citations as the weight to the corresponding edge in the unweighted citation network.
4) We create different graph based paper recommendation systems on the weighted network.

A. Paper Content Extraction

For a given unweighted citation network with metadata of nodes, we collect PDF files of papers, and extract the text from the PDF files using Apache PDF Box and PyPDF2.

Parsing PDFs is difficult and error-prone. Major sources of error include papers written in two-column format, lossy translation from PDF to plain-text, characters with diacritical marks, and human error.

Papers written in two-column format were read as single-column papers. The PDF parsers would read the first line of Figure 2 as "REFERENCES 16 Gerard Salton and C.S." and so on.

Parsing PDFs is lossy. Super-script and sub-script text were read as normal text. Font sizes were forgotten upon translation to plain text, making it impossible to distinguish footnotes from the body of the papers. Sometimes the PDF reader would capture text from graphs, images, and figures.

PDF parsers also struggle to handle characters with diacritical marks, such as "ö" and "á." The parser replaces these characters with a series of extra symbols, letters, and numbers. This type of error affects the authors' names, paper title, and publication venue.

Finally, human error makes it difficult to scrape citations and references from the text. Authors make mistakes when writing references or making citations.

In order to handle the various errors introduced from parsing the PDFs, we normalize the text. The normalization process consists of replacing non-letter and non-numeric characters with a space, setting all letters to lower-case, and removing newlines.

B. Citation Weight Extraction

After normalizing the text, we use the metadata to create a series of regular expressions to extract each occurrence and the context of a citation and reference. We use the term "reference" to mean the citation in the reference section of a paper and the term "citation" as both a reference as well as an instance in the body of the paper where the author cites another paper. The term "hit" refers to an instance where the regular expression matches a piece of text in the paper. Citation area or "chunk" will refer to an area of text including n-characters surrounding either side of a hit.

There are a wide variety of citation styles, and we use flexible regular expressions to capture as many citations as possible. Our implementation works with 90% accuracy for a sample of 10 random papers. The 10% error was due to human error. These techniques could be improved to cover more cases not included in the small sample.

From the given citation network, we know the citing paper, the cited paper, and the metadata for both papers. This information allows us to create regular expressions to search for common patterns using authors' names, publication year, and title.

After finding all of the potential citations based on authors' last names using regular expression matching, it is important to find citations using numeric indicators (e.g., [1]). To find the reference and potentially a linked number, we use the longest word in the title as a simple heuristic to find all potential references. For each potential reference found, we take a chunk surrounding the hit's location in the text. This chunk is a potential reference.

Given this list of potential references, we measure how similar the cited paper's metadata is to each chunk in the list. We measure similarity by creating a target set of words based on the paper's metadata. We convert each chunk into a set of words, and treat the chunk with the greatest intersection with the target set as the reference.

If this reference is linked to a number, the number will typically be immediately left of the author's name. For each word in the chunk, we extract any groups of numbers. We then pick and remember the right-most number. If the current word equals the first author's last name, the current number is the reference's numeric indicator.

We use this number to find occurrences of citations made numerically by using another regular expression and checking each hit's surroundings for brackets ("[", "]"). The pattern is flexible enough to find matches occurring in a list of references (e.g., [1,5,3,9]).

Finally, we have a complete list of hits, where each hit is a citation. We remove all duplicate citations and reference, ending with the final weight. This process was done to both versions of text (PyPDF2 and Apache PDF Box). If there is a difference between the two versions, we choose the higher weight to add it to the citation network.

The citation extraction works well only for two types of citations: those using numbers in brackets and authors' last names. The citation weight finder could not work with citation styles using super scripted numbers because the PDF parsers read super scripted numbers as regular numbers. Furthermore, the weight finder highly depends on an accurate first author's last name.

C. Recommendation Systems on Weighted Citation Network

We construct the paper recommendation system on our weighted citation network using different models created for this aim. While they work on an unweighted citation network, we modify them to work on weighted network.

1) Weighted CCIDF & BC: Co-citation Inverse Document Frequency (CCIDF) and Bibliographic Coupling (BC) are two main citation-based techniques that measure the similarity of two different articles based on their relationships with other articles. CCIDF is based on common citations to measure the relatedness between paper, and BC is based on the number of shared citations. CCIDF and BC are designed to work on unweighted graphs. CCIDF looks at papers with co-citations, and then takes the inverse frequency of documents with that citation [3]. The reason it uses the inverse frequency is to give shared uncommon citations a higher weight. Simultaneously, since seminal papers are cited by a diverse range of papers, co-citations of frequently cited papers are discounted. This increases the likelihood that CCIDF provides relevant recommendations.

To calculate the similarity between two papers, p_i and p_j, we identify all articles that are referenced by p_i and p_j as a set P_{coref}. Then similarity is defined as follows:

$$Sim(P_x, P_y) = \sum_{i=1}^{n} \frac{1}{NDL(P_{co-ref[i]})} \tag{1}$$

where $NDL(P_{co-ref[i]})$ is the number of directed links from other nodes to the node $P_{coref[i]}$ in the citation network.

Similar to CCIDF, BC uses the bibliographic coupling network extracted from citation networks, where two papers are linked if they both cite same article [34].

In order to test the performance of CCIDF and BC on weighted graphs, we modify the algorithms. In the weighted graph, edge weights are equal to the number of times the citing paper cites the cited one. Instead of using the inverse frequency of common citations, we used the edge weight divided by the total number of citations the citing paper makes.

Then weighted similarity is defined as follows:

$$Sim(P_x, P_y) = \sum_{i=1}^{n} \frac{w_{x,i}}{SWDL(P_{co-ref[i]})} \tag{2}$$

where $SWDL(P_{co-ref[i]})$ is the sum of the weight of directed links from other nodes to the node $P_{coref[i]}$ in the citation network.

Similarly with BC, instead of using the total number of shared references, we use total weights which are the total number of times a potential recommendation cites a shared reference.

Given the toy unweighted citation network in Figure 3a(a), we consider the input paper as the query paper and compute the CCIDF and BC scores for it. According to these scores, Paper B and Paper C are most similar papers to input paper. After we convert it to weighted citation network, we compute weighted scores on it and Paper B and Paper C are still most similar to the input paper. Notice, however, that the weighted CCIDF ranks Paper C above Paper B.

2) RWR: Random Walk with Restart [35] determines the similarity of each node using a ranking vector. The rank of a single node is determined by the weights of edges leading to it.

$$R(p_i) = \frac{1-\alpha}{N} + \alpha \sum_{p_j \in A_{(p_i)}} R(p_j) P(p_j, p_i) \tag{3}$$

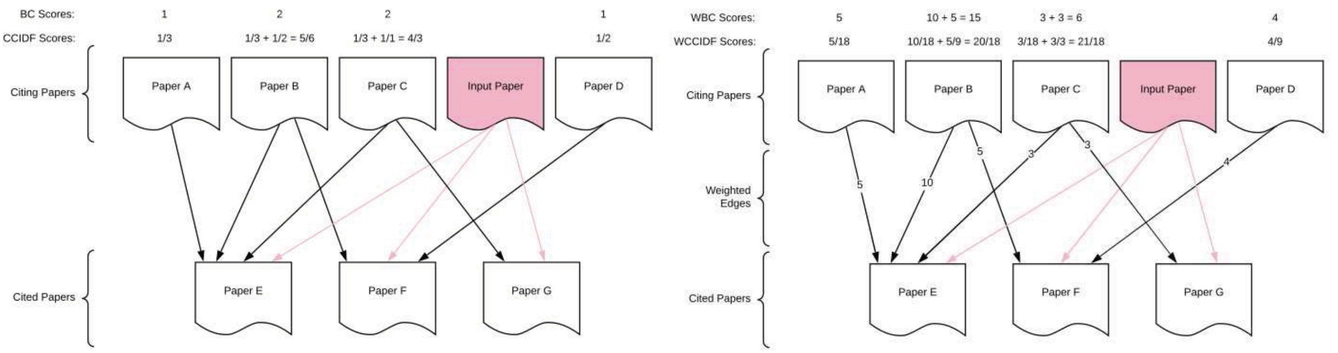
In Equation 3, R is a ranking vector and $R(p_i)$ is the rank score of some node p_i. $A(p_i)$ is the set of nodes adjacent to p_i. α is the damping factor and $P(p_j, p_i)$ is the probability of moving from node p_j to p_i. Equation 3 captures the movements of an entity randomly walking across the network. With each step the walker takes, it has a chance to move to a neighbor node with probability α and a chance to restart to its origin node with probability $1 - \alpha$.

We can find the rank score vector for all of the nodes in the network iteratively as defined by Equation 4.

$$R^{(t+1)} = \alpha S R^{(t)} + (1-\alpha) q \tag{4}$$

The rank score vector at step t is $R^{(t)}$. q is a vector of (0, ... , 1, ... , 0). The element in q equal to 1 is the input paper, and all other elements are 0. S is a transfer matrix of probabilities of moving from one node to the next ($P(p_j, p_i)$ in Equation 3).

In the original model, the probability transfer is defined as the inverse of the total number of in-links of paper i. However, using citation weights, the transfer probability is defined as the number of times paper j cites paper i divided by the total



(a) Original Scores

(b) Weighted Scores

Figure 3: Toy Citation Network

number of citations p_j makes as given in Equation 5. The edge connecting paper j to paper i is $L(p_j, p_i)$ and $W(l)$ is the weight of some link l. $OL(p_j)$ is a collection of paper j's out edges.

$$P(p_j, p_i) = \frac{W(L(p_j, p_i))}{\sum\limits_{l \in OL(p_j)} W(l)} \tag{5}$$

IV. Experiment and Results

We perform experimental studies to evaluate the effectiveness of our paper recommendation system in a real-world network. We first provide an overview of the datasets used for experiments. We further show results for the weighted and unweighted network for 3 different measures.

A. Dataset

In our experiment, we use the ACL Anthology Network (AAN) released in 2014 [36]. The AAN is an unweighted citation network consisting of 23,775 nodes, 124,842 edges, and an average degree of 10.5. The metadata of each node includes a paper ID, authors, title, venue, and publication year. The citation network contains unweighted edges connecting papers using their paper IDs. After running the citation extraction process, the final network contains 2082 nodes and 8194 edges with an average degree of 7.87.

B. Evaluation

We find rankings of paper relevance for a given input paper using 3 different similarity measures: (1) BC, (2) CCIDF, and (3) RWR. The top-k ranked papers are returned as recommendations according to their similarity scores.

We compare the results of each measure (CCIDF, BC, and RWR) on the unweighted and weighted citation networks. We use three commonly-used measures to evaluate the change in recommendation systems' performance: precision (P), recall (R), and F_1-score (F). [3] We calculate metrics using the *All-But-One* method.

Given a target node, p, we randomly select 10% of its neighbors as a test set, T. If the node's degree is less than 10, we select 1 neighbor. Then, we remove the edges linking the target node and each node in the test set. Finally, we run the recommendation system and select the top-k set of recommended papers, D. We examine how many test nodes occur in the result list after removing the citation links D. Precision measures how well the test set covers the recommendation set. Recall measures how well the recommendation set covers the test set. F_1-score captures the relationship between precision and recall. Equations for these measures are given below as R is for recall, P is for precision, and F_1 is for F_1 score;

$$R = \frac{|D \cap T|}{|T|} \quad P = \frac{|D \cap T|}{|D|} \quad F_1 = \frac{2 * P * R}{P + R} \tag{6}$$

As the recommendation list increases, we expect precision to decrease and recall to increase. For each recommendation algorithm, we select 200 random nodes and calculated metrics using the *All-But-One* method on both the weighted and unweighted networks. We find the metrics from a recommendation set ranging from one to thirty in increments of two papers, and repeat this process ten times to generate the data.

C. Results

In this section, we compare the algorithms (1) BC, (2) CCIDF, and (3) RWR using weighted and unweighted networks for different k values. The results are available in Figures 4, 5 and 6. The y-axis is the metric and the x-axis is the length of the recommendation list, k.

For the BC algorithm, as we see in Figure 4, both methods get a higher recall for bigger k values. For most k values, recall for weighted network is higher than unweighted network. While precision on small k for unweighted network is very low, weighted network has higher precision for all k values. While for small k values, weighted network has higher F_1 than unweighted network, for bigger k values, there is no significant difference between them.

For the CCIDF algorithm, as we see in Figure 5, the recall rates are getting higher as k increases. As a comparison, the weighted network has mostly better recall rates than the unweighted graph. For precision, although the unweighted

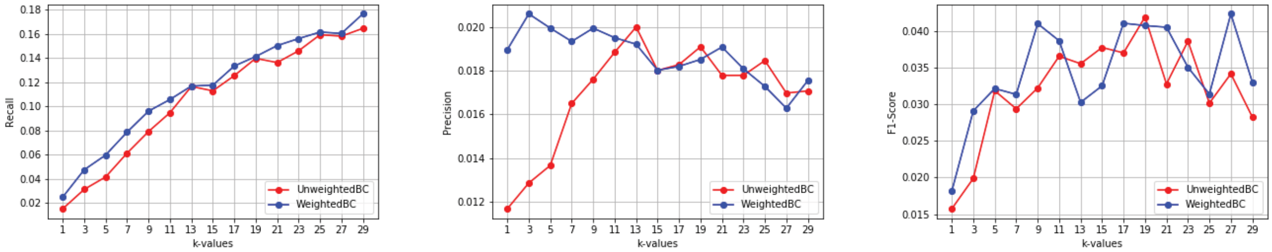


Figure 4: Recall, Precision, and F_1 Scores for the BC Algorithm

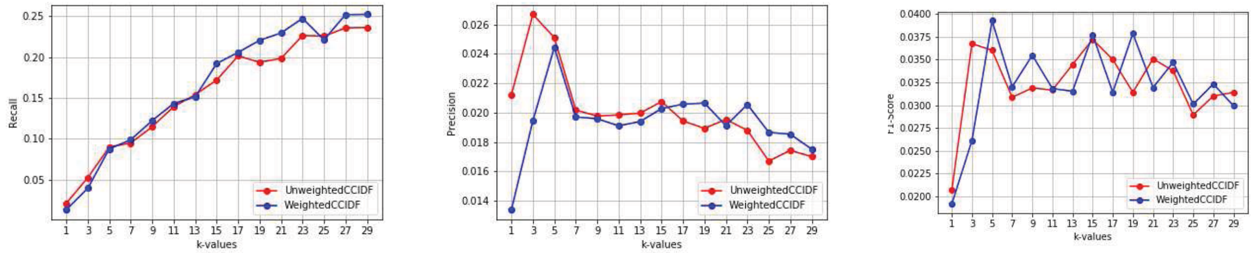


Figure 5: Recall, Precision, and F_1 Scores for the CCIDF Algorithm

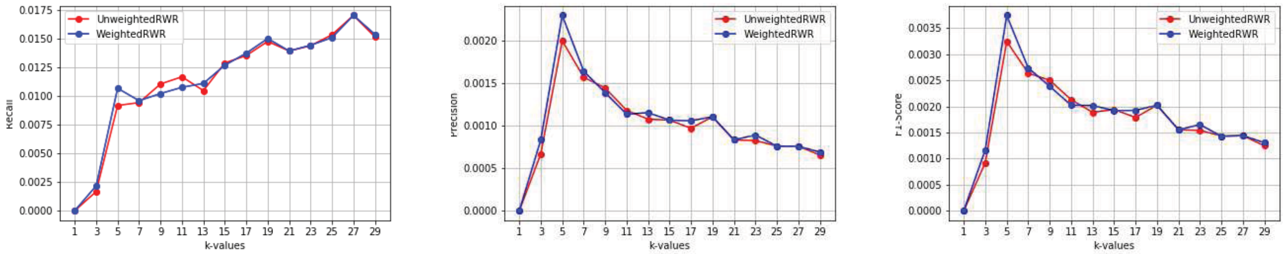


Figure 6: Recall, Precision, and F_1 Scores for the RWR algorithm

network has better rates for small k values, both methods are comparable for bigger k values.

For the RWR algorithm, as we see in Figure 6, both methods have similar recall, precision, and F_1 scores. However, for $k = 5$, the weighted network has significantly better results than the unweighted network.

After evaluating each recommendation system on the unweighted and weighted networks, we can see that there is a marginal increase in accuracy using weighted networks. Accuracy using weighted networks is approximately 1% higher than unweighted networks. The results suggest that using relevance of citation relations contributes to finding more relevant papers.

V. Conclusions and Future Work

While the academic community publishes millions of research articles each year, the literature search is becoming a more challenging and time-consuming task. In this paper we propose a method that considers the difference between cited papers. We extract citations from papers to create a weighted paper citation network. Using weighted paper citation networks, we determine that common graph-based recommendation algorithms marginally increase in performance. It is hard to say how this research fits into broader context. The academic paper recommendation system is a growing field, and there is some evidence that metrics-based evaluations are not good indicators of recommendation system performance because they ignore all of the human factors involved in recommendation systems [3]. It is possible that the weights impact user satisfaction in online experiments more significantly.

Additionally, there is no established baseline for paper recommendation system comparison [3]. We hope that our findings help researchers understand the impact of citation-based weights on graph-based recommendation systems, and consider incorporating this citation relation into other recommendation algorithms. Further work includes improving the citation extraction algorithm to make a larger citation network, and re-evaluating our method on that larger network. We will be publishing all codes used in this project at https://github.com/wntanner/PaperRecommendations for public use.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1659645. This work was done in part through the Big Data Analytics REU program at Oklahoma State University. We would also like to thank Sophia Tanner for her patiently helping with proofreading and editing.

References

[1] R. Johnson, M. Mabe, and A. Watkins, "The stm report: An overview of scholarly and scientific publishing." International Association of Scientific, Technical and Medical Publishers, 2018.

[2] O. Kktun, E. Saule, K. Kaya, and U. Catalyurek, "Recommendation on academic networks using direction aware citation analysis," 05 2012.

[3] J. Beel, C. Breitinger, and S. Langer, "Evaluating the cc-idf citation-weighting scheme: How effectively can inverse document frequency (idf) be applied to references?" *iConference 2017 Proceedings*, 2017.

[4] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.

[5] J. Lin and W. J. Wilbur, "Pubmed related articles: a probabilistic topic-based model for content similarity," *BMC bioinformatics*, vol. 8, no. 1, p. 423, 2007.

[6] M. D. Ekstrand, P. Kannan, J. A. Stemper, J. T. Butler, J. A. Konstan, and J. T. Riedl, "Automatically building research reading lists," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 159–166.

[7] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed-membership models of scientific publications," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5220–5227, 2004.

[8] K. Jack, "Mendeley: recommendation systems for academic literature," *Presentation at Technical University of Graz (TUG)*, 2012.

[9] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt, "Capturing knowledge of user preferences: ontologies in recommender systems," in *Proceedings of the 1st international conference on Knowledge capture*. ACM, 2001, pp. 100–107.

[10] S. Kataria, P. Mitra, and S. Bhatia, "Utilizing context in generative bayesian models for linked corpus," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[11] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[12] S. Huang, G.-R. Xue, B.-Y. Zhang, Z. Chen, Y. Yu, and W.-Y. Ma, "Tssp: A reinforcement algorithm to find related papers," in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 2004, pp. 117–123.

[13] C. Nascimento, A. H. Laender, A. S. da Silva, and M. A. Gonçalves, "A source independent framework for research paper recommendation," in *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*. ACM, 2011, pp. 297–306.

[14] A. Naak, H. Hage, and E. Aimeur, "A multi-criteria collaborative filtering approach for research paper recommendation in papyres," in *International Conference on E-Technologies*. Springer, 2009, pp. 25–39.

[15] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, "On the recommending of citations for research papers," in *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. ACM, 2002, pp. 116–125.

[16] C. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang, "Cares: a ranking-oriented cadal recommender system," in *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2009, pp. 203–212.

[17] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl, "Enhancing digital libraries with techlens+," in *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2004, pp. 228–236.

[18] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 473–480.

[19] N. Agarwal, E. Haque, H. Liu, and L. Parsons, "Research paper recommender systems: A subspace clustering approach," in *International Conference on Web-Age Information Management*. Springer, 2005, pp. 475–491.

[20] Y. Liang, Q. Li, and T. Qian, "Finding relevant papers based on citation relations," in *Proceedings of the 12th international conference on Web-age information management*. Springer, 2011, pp. 403–414.

[21] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *Journal of the American Society for information Science*, vol. 24, no. 4, pp. 265–269, 1973.

[22] M. M. Kessler, "Bibliographic coupling between scientific papers," *American documentation*, vol. 14, no. 1, pp. 10–25, 1963.

[23] B. Gipp and J. Beel, "Citation proximity analysis (cpa): A new approach for identifying related work based on co-citation analysis," in *ISSI09: 12th International Conference on Scientometrics and Informetrics*, 2009, pp. 571–575.

[24] B. Gipp, J. Beel, and C. Hentschel, "Scienstein: A research paper recommender system," in *Proceedings of the international conference on emerging trends in computing (icetic09)*, 2009, pp. 309–315.

[25] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine learning*, vol. 81, no. 1, pp. 53–67, 2010.

[26] M. Baez, D. Mirylenka, and C. Parra, "Understanding and supporting search for scholarly knowledge," *Proceeding of the 7th European Computer Science Summit*, pp. 1–8, 2011.

[27] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles, "Learning multiple graphs for document recommendations," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 141–150.

[28] A. Woodruff, R. Gossweiler, J. Pitkow, E. H. Chi, and S. K. Card, "Enhancing a digital book with a reading recommender," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2000, pp. 153–160.

[29] A. D. Wade, "Overview of microsoft academic graph," *Alonso et al.[2]*, p. 8, 2015.

[30] D. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara, "The acl anthology network corpus," *Language Resources and Evaluation*, pp. 1–26, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10579-012-9211-2

[31] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[32] X. Wan and F. Liu, "Are all literature citations equally important? automatic citation strength estimation and its applications," *Journal of the Association for Information Science and Technology*, vol. 65, no. 9, pp. 1929–1938, 2014.

[33] "CiteSeerX about citeseerx," http://csxstatic.ist.psu.edu/home, accessed: 2019-08-28.

[34] L. Egghe and R. Rousseau, "Co-citation, bibliographic coupling and a characterization of lattice citation networks," *Scientometrics*, vol. 55, 11 2002.

[35] Z. Chen, F. Xia, H. Jiang, H. Liu, and J. Zhang, "Aver: Random walk based academic venue recommendation," 05 2015, pp. 579–584.

[36] D. R. Radev, P. Muthukrishnan, and V. Qazvinian, "The ACL anthology network corpus," in *"Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries"*, "Singapore", "2009".