Where to begin? Using Network Analytics for the Recommendation of Scientific Papers

Laura Steinert, Irene-Angelica Chounta, and H. Ulrich Hoppe

University of Duisburg-Essen, Lotharstr. 63, Duisburg, Germany

Abstract. This paper proposes a network analytic approach for scientific paper recommendations to researchers and academic learners. The proposed approach makes use of the similarity between citing and cited papers to eliminate irrelevant citations. This is achieved by combining both content-related and network-based similarities. The process of selecting recommendations is inspired by the ways researchers adopt in literature search, i.e. traversing certain paths in a citation network by omitting others. In this paper, we present the application of the newly devised algorithm to provide paper recommendations. To evaluate the results, we conducted a study in which human raters evaluated the paper recommendations and the ratings were compared to the results of other network analytic algorithms (such as Main Path Analysis and Modularity Clustering) and a well known recommendation algorithm (Collaborative Filtering). The evaluation shows that the newly devised algorithm yields good results comparable to those generated by Collaborative Filtering and exceeds those of the other network analytic algorithms.

1 Introduction

One task any researcher has to face quite often, regardless the research field, is literature research. Whether a scientist is new to a field or exploring a new idea, the time consuming task of skimming seemingly hundreds of papers lies ahead in order to find the few precious papers that are of interest. "Nowadays, [...] there are too many articles to read [...], and as a result, intelligent recommender systems [...] play a crucial role in recommending articles." [5, p. 1830]. By providing a user with literature recommendations he or she is also indirectly presented scientific communities that may fit his or her interests. In this way, we aim to support a new researcher to place himself within a community and identify potential useful collaborations and connections.

A strategy applied by many researchers when they look for relevant papers is to pursue a chain of citations. However, not all citations are followed because not all citations are equally important to the topic of interest. This could be expressed by weighing the citations differently. Notwithstanding, in previous research citations were usually considered to be equal in citation analysis [5]. Garfield [7] lists 15 reasons to cite a document, additionally saying that these are only some of the possible reasons. Accordingly, a "purely quantitative citation analysis" is heavily criticized [18, p. 103]. Hence, whenever a citation network – i.e. a graph where nodes are papers and edges represent citations among them – should be used to generate reading recommendations, only citations that contribute to the main topic should be taken into account. Therefore, a method is needed to weight citations in a citation network in order to reflect the relevance of citations. This is also pointed out by Liu and Lu [11].

This paper proposes a method to generate reading recommendations based on citation networks that additionally takes the relevance of citations into account. This relevance is calculated by combining content-related similarity measures with similarities derived from the structure of the citation network. It is assumed that a cited paper that is similar in content and citation patterns to the paper citing it, is a relevant citation. The innovative key idea is to consider the paths of citations humans follow to generate reading recommendations. This idea was originally inspired by Main Path Analysis [9] [2]. Hence, the new approach is grounded in network analytic methods. By adapting the human approach to find relevant literature, the new method is more intuitive to understand than existing methods. Moreover, we believe that by using the structure of the citation network to generate recommendations, the quality of recommendations can be improved compared to algorithms that consider all papers as recommendations.

The next section of the paper provides an overview of related work. Afterwards, a novel algorithm to recommend papers is presented alongside two known algorithms that are also adapted for this task. In Section 4, we present a study where the results of the aforementioned algorithms were evaluated. The results are discussed in Section 5. The paper ends with a conclusion and a prospect of future work.

2 Related Work

The need to find literature on a given topic is closely intertwined with scientific research. As such, many tools came up over the preceding decades to aid researchers in that quest. Nowadays, many platforms exist that allow researchers to traverse citation networks easily – e.g. CiteSeerX[1]. Although citation networks enable a user to find literature on a given topic, not all cited papers are relevant. This is because not all papers linked via a citation are related in topic as has been mentioned previously.

Pohl et al. [14] recommend papers based on a co-download measure of papers to indicate topic similarity / relatedness. Others rely on textual similarity alone to recommend papers. For example Lee et al. [10] use all papers published by a querying user and compare the abstracts and titles of these to a corpus of papers. They use a vector space model and cosine similarity to calculate the similarities between papers. The papers most similar to the querying user's papers are then recommended.

[1] `http://citeseerx.ist.psu.edu`, as seen on March 9th 2015

Other recommender systems for scientific papers make use both of the textual information as well as the citation network. Sugiyama and Kan [17] use a user's papers as a profile against which the similarities of possible recommendations are calculated. They construct a feature vector for each paper of the user by calculating term-frequencies. Additionally, the feature vector encodes the similarity of the paper to its citing and cited papers. Hence, they partly make use of the information given by the citation network. Recommendation candidate papers – for which similar feature vectors are constructed – are then compared to the feature vectors of the user and the most similar papers are recommended.

Torres et al. [19] generate reading recommendations for scientific papers by combining collaborative filtering (CF) and content-based filtering (CBF). Both methods can be run in parallel or sequentially. As input a user has to select one paper that is used to generate his profile of interest. For the content-based filtering the approach uses *term frequency-inverse document frequency* (tf-idf) on the text of papers, which are subjected to stopword removal and stemming beforehand. The CF approach is taken from McNee et al. [12], where k-nearest neighbour is used. A matrix where rows are papers and columns are citations is generated. An entry in the matrix denotes whether a paper has cited another paper. By computing the cosine similarity between the rows of the matrix – i.e. the citation patterns of papers – the CF algorithm recommends papers with a high similarity to the paper selected by the user. Overall they test a variety of combinations of the two modes. If run in parallel, the results of both algorithms are merged to generate recommendations with papers highly recommended by both algorithms ranking higher in the overall results. If run sequentially the output of one algorithm is the input of the other. In [19] variants are presented where any one of the two is the first algorithm.

Closely related to recommending papers to read is the task of recommending papers to cite, which is for e.g. done by Strohman et al. [16]. In their approach both the citation network as well as textual similarities are combined. As an input the user has to enter a text about the topic he is writing about. To retrieve a set of candidate papers, they compute the text similarity of the input text to over one million papers in a database and return the 100 most similar papers as the set R. However, they note that in the database used – Rexa – only approximately 10 percent of the entries contain full text information. In a second step they add all papers to R that any paper in R cites. Adding additional papers to that set, e.g. by adding the papers cited by the newly entered entries, does not improve the results. They report that R usually contains 1000 to 3000 documents and contains 90 percent of the papers researchers actually cite given the input text. They then rank the papers in R according to a mix of six features – publication year, text similarity, co-citation coupling, same author, Katz graph distance and citation count – and recommend those ranked highest.

A network analytic algorithm not intended for recommending papers but that might very well be suited for this task is the so called *Main Path Analysis* (MP). This algorithm discovers the path along which information is dispersed among scientific publications. As such the found main path can yield information about the most influential papers and therefore can serve as a reading suggestion for scientists new to this field. Originally proposed by Hummon and Doreian [9], the first step of the algorithm was considerably improved by Batagelj [2]. For a given directed acyclic graph a single source vertex and a single sink vertex are added and connected to the previously present sources or sinks. A source is defined as a vertex with an indegree of zero, whereas a sink has an outdegree of zero.

The MP algorithm then is a two stepped procedure that first calculates a flow for the graph. A flow is a function that maps every edge to a natural number and all vertices (except the source and the sink) fulfill Kirchoff's current law, i.e. the sum of the flows of all incoming edges equals the sum of the flows of all outgoing edges. In the second step the (local) main path is derived from the calculated SPC weights. The algorithm starts in the source vertex and always selects the outgoing edge with the highest SPC weight. If two or more edges exist that have the highest weight, the main path splits and continues along all of these edges. Upon reaching the sink the main path ends.

3 Method of the study

A scientist searching for literature usually traverses a citation network starting from a single paper and following a subset of all possible citations. As mentioned before, this is because not all citations are relevant. However, although some of the approaches mentioned in Section (2) use the structure of the citation network to determine the similarity between papers, none make use of the network to actually generate the set of candidates. Most approaches look at *all* papers, calculate their similarity and recommend the most similar. But why should an algorithm not mimic the human approach? One might argue that a researcher would gladly look at all papers in the citation network if only he or she had the time. Moreover, a researcher often does not have access to the whole citation network but only sees the literature at the end of each paper. However, maybe there is more to the human strategy of finding relevant literature. Therefore, we propose an algorithm that mimics the way humans search for literature (cf. Section 3.1). Furthermore, we adapt other algorithms that were not initially designed for giving reading recommendations but that exploit the structure of the underlying graph (cf. Sections 3.2 and 3.3). All three algorithms are network analytic approaches to the problem.

Figure 1 illustrates the overall process to recommend literature. First of all, the citation network has to be extracted. Afterwards, content-related and graph-based similarities are calculated and combined. Finally, the literature recommendations can be generated. These steps are described in the following subsections.

Extracting the Citation Network

In order to receive reading recommendations a user has to name one paper that covers the topic he or she wants to explore further. The citation network used

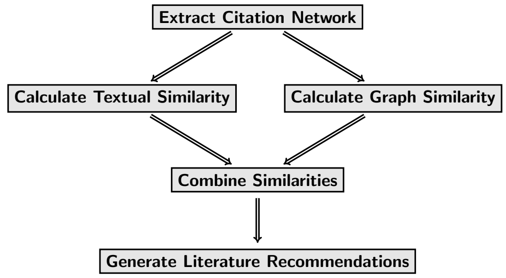

Fig. 1. The overall process to recommend literature

in this paper was gathered by extracting it from CiteSeerX[2] by starting from the paper the user provided as a seed. From there, the citations and inverse citations – i.e. the *cited by* relations – were crawled up to a given depth. Papers were added as vertices only if the abstract was provided on CiteSeerX and if it was written in English. Additionally, the title was extracted. Citations were added as edges to the extracted graph. The edges always point from the citing to the cited papers, i.e. backwards in time.

Calculating the Similarity

In our approach textual similarities are combined with similarities based on the structure of the citation network to generate the *topic-structure similarity* (tss), since Strohman et al. [16] report that using such a combination outperforms recommendations based on only one of the similarity measures. This was also done by He et al. [8]. As a measure based on the structure of the citation network the strength of the co-citation coupling is used.

The co-citation coupling was defined by Small [15] and is a subject similarity indicator. It measures how often two articles are cited together in relation to their overall individual citations. The strongest co-citation couples seem to be papers that are also directly liked by citation, yet this does not apply to all strongly coupled papers. It is believed to be a better subject similarity indicator than bibliographic coupling.

The strength $CoCitS$ of the coupling is calculated using Equation (1) suggested by Garfield [6]. In this equation, $coCit(u, v)$ stands for the number of co-citations of u and v, whereas $cit(u)$ gives the overall number of citations of text u. Since every co-citation is counted twice in $cit(u) + cit(v)$, the number of co-citations has to be subtracted in the denominator from the overall number of citations.

$$CoCitS(u,v) = \frac{coCit(u,v)}{cit(u) + cit(v) - coCit(u,v)} \tag{1}$$

[2] `http://citeseerx.ist.psu.edu`, as seen on 9th March 2015

The topic-structure similarity was only calculated for papers citing each other, i.e. that are directly connected in the citation network. This similarity was encoded as an edge weight. In order to find the linguistic algorithms best suited for calculating the content-related similarity, a small evaluation was conducted using different text similarity algorithms in combination with various key phrase extraction and preprocessing algorithms, e.g. lemmatizing. A set of papers from three different scientific domains was taken and the similarity between every pair calculated. Afterwards, modularity clustering [4] was applied. The results were then scored depending on how well the generated clusters resembled the three expected clusters. The combination of *Co-occurrence Graph key phrase extraction* and *Greedy-String-Tiling* (GST) yielded very good results with a very good runtime behaviour. Hence, it was selected to calculate the content-related similarity used in this paper.

The algorithm called *Co-occurrence Graph key phrase extraction* is a variant of the *TextRank* algorithm by Mihalcea and Tarau [13]. It is an unsupervised method to extract key phrases from natural language texts. Unsupervised means that the system does not need to be trained on a corpus. The algorithm first constructs a graph from a given natural language text. The vertices are sequences of words from the texts. An edge is established between two vertices if the corresponding word sequences co-occur within a section of the text that is maximally N words long. N is also referred to as the *window size*. However, words occurring in the same window have to belong to the same sentence. Additionally, the words making up the vertices can be further filtered by word class. In this variant only nouns are permitted. The minimal length of key phrases is two and the maximum size is set to four. The window size is set to two. After the graph has been created, a score for every vertex is calculated, encoding how well it can be used as a key phrase for the text. In this work the implementation provided by *DKPro Keyphrases*[3] is used.

The Greedy-String-Tiling algorithm as described by Wise [20] is an algorithm to calculate content-related similarity by detecting equal substrings in two texts. A pair of matching substrings is called a *tile*. For this only substrings of a minimal length of three are considered. Moreover, tiles never overlap.

The algorithm seeks to maximize the tiling of the two texts, i.e. it tries to maximize the number of tokens belonging to tiles. However, since this problem is probably NP-hard, if multiple tilings are possible GST is set to prefer larger tiles. This increases the chances of finding significantly similar passages versus mere chance similarities as produced by smaller tiles. The similarity is then the percentage of the first text that is covered by tiles. The implementation used was provided by the DKPro Similarity[4] library.

In order to calculate the content-related similarity of two papers in our approach the title and abstract information was subjected to key phrase extraction using *Co-occurrence Graph key phrase extraction*. Here all extracted key phrases regardless of their weight were considered. The content-related similarity be-

[3] `http://code.google.com/p/dkpro-keyphrases/`, as seen on March 9th 2015
[4] `https://code.google.com/p/dkpro-similarity-asl/`, as seen on March 13th 2015

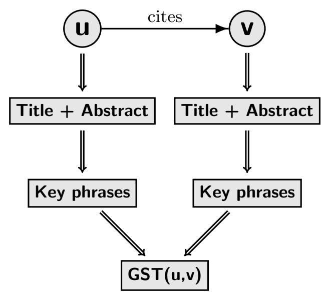

Fig. 2. The calculation of the content-related similarity of two papers u and v

tween the key phrases of both papers was then calculated using GST. This process is illustrated in Figure 2. Both the content-related similarity as well as the co-citation coupling strength were then summed up, resulting in a topic-structure similarity $tss(u, v) \in [0; 2]$ for two vertices u and v. This is also shown by Equation (2).

$$tss(u, v) = CoCitS(u, v) + GST(u, v) \tag{2}$$

Once the citation network has been extracted and the edge weights have been calculated, a reading recommendation can be calculated by using one of the following three algorithms.

3.1 Finding Highly Connected Components

A citation network is a directed acyclic graph. Hence, strongly connected components cannot exist. Therefore, a weakly connected component will be referred to as *component* only.

By construction the citation network is a single component. However, edges with a low weight – i.e. low topic-structure similarity – are of little interest. By repeatedly deleting the edges with the lowest weights the graph will be segmented into different components until eventually only isolated vertices exist. However, assuming that papers covering the same topic are highly connected among one another with high edge weights they should be found in the same component up until very late in the process. In order to generate reading recommendations one could therefore look at this largest component and give its set of vertices as recommendations. The problematic part is up to which edge weight edges shall be deleted.

Let θ be the threshold such that all edges with an edge weight below it are deleted. If one only considers the largest component, i.e. the component with the most vertices, for any θ, the sum of edge weights and the average edge weight describe functions nearly monotonic in nature. With an increasing θ, the sum of edge weights in the largest component decreases since all edge weights are greater or equal to zero and for $\theta = 0$ the highest number of edges – all edges

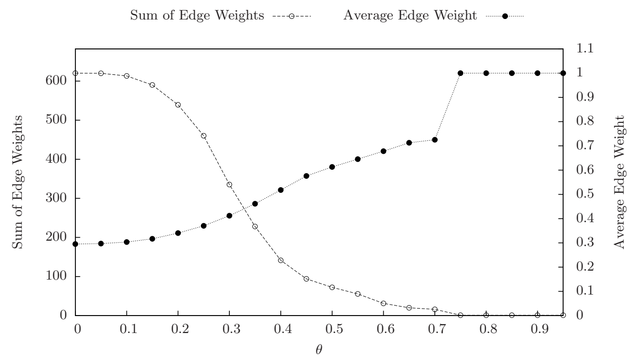

Fig. 3. The sum and the average edge weight for the largest component of a real citation network - generated by taking a paper by Fortunato as seed

– are in the largest component. However, as more and more edges with lower weights are deleted, the largest component will have an increasing average edge weight. This is depicted in Figure 3 for a real world citation network. Here, the topic-structure similarity edge weights were normalized beforehand. The seed of this citation network is a paper by Fortunato[5]. The depth of the retrieved citation network was three, i.e. if edge direction is ignored, any vertex in the graph is connected to the seed vertex by a path of maximally length three. It contains 1451 vertices.

In order to find the θ that determines the largest component that is returned as a reading recommendation, two variants are possible. The first assumes that since the seed paper concerns the topic of interest, the highest θ for which the largest component still contains the seed paper should be taken. However, this is the case where all vertices are isolated, i.e. the seed vertex forms a component of its own. Thus, a constraint should be added that only largest components that contain more than $minSize$ vertices are considered. For all evaluations described in this paper $minSize$ was set to one. This method is described in Algorithm 1 and will be called *LC Seed*.

The second variant seeks to find the optimal θ thus that all topic-structurally not very similar papers have been disconnected from the largest component but that all highly similar papers are still connected. Inspired by the diagram depicted in Figure 3 we introduce an intersection heuristic. The θ for which the function of the average edge weight and the function of the sum of edge weights intersect is selected and the respective vertices in the largest component returned as reading recommendations. This could be approximated by increasing the θ discretely and finding the θ for which the two functions differ the least. However, before the two functions can be compared, their ranges need to be equalized. First of all, since the topic-structure similarity lies in $[0; 2]$ it can be normalized

[5] Fortunato, S.: Community detection in graphs. Physics Reports 786(3), 75-174 (2010)

Algorithm 1 LC Seed

Require: weighted Citation Network *graph*
  N = $\emptyset$
  for $\theta \leftarrow 0$ **to** 1 **do**
    Delete edges with weight $< \theta$
    C = largest components
    for $c \in C$ **do**
      if $seed \in V(c)$ **then**
        if $|V(c)| > minSize$ **then**
          N = V(c)
        end if
      end if
    end for
  end for
  return N

Algorithm 2 LC Intersection

Require: weighted Citation Network *graph*
  min = $|V(graph)|$
  N = $\emptyset$
  maxSum = SumOfWeights(graph)
  for $\theta \leftarrow 0$ **to** 1 **do**
    Delete edges with weight $< \theta$
    C = largest components
    c' = $\emptyset$
    for $c \in C$ **do**
      c' = c
      if $seed \in V(c)$ **then**
        break {If there are multiple largest components, prefer that with the seed vertex}
      end if
    end for
    $\Delta = \left|avgWeight(c') \cdot maxSum - sumOfWeights(c')\right|$
    if $\Delta < min$ **then**
      $min = \Delta$
      $N = V(c')$
    end if
  end for
  return N

by dividing all weights by the highest occurring topic-structure similarity. Then the average edge weight lies within $[0;1]$. By multiplying it with the maximal sum of edge weights – which is the sum of edge weights of the whole citation network – the ranges of the two functions can be equalized. This approach is described by Algorithm 2 and will be called *LC Intersection*.

Both *LC Seed* and *LC Intersection* can be run using the topic-structure similarity as an edge weight. However, by multiplying the topic-structure similarity with the SPC weights generated by the first step of the Main Path Analysis and using that as edge weights the notion of the evolution of a scientific field can also be taken into consideration. These variants will be called *LC Seed SPC*topSim* and *LC Intersection SPC*topSim*.

3.2 Weighted Main Path

This normal calculation of the main path is only based on the citation network but not on the content-based similarity of the papers. By combining the SPC weights with the topic-structure similarity via a multiplication – as was done in [3] – and finding the main path on these altered edge weights this problem can be overcome. Hence, this weighted Main Path contains only papers that are important concerning the SPC weights where adjacent papers are additionally topically and structurally similar. Thus, this weighted main path might contain good reading recommendations. But one problem remains: The weighted main path might not contain the seed paper and might thus be concerned with a different topic than the one for which recommendations are sought. This might be overcome by adding a restriction that the main path must contain the seed paper. This algorithm will be called *weighted Main Path* or *WMP*.

3.3 Modularity Clustering

Papers concerning the same topic should have a high topic-structure similarity. Moreover, it is likely that they are well connected among one another since papers on the same topic might cite each other. Hence, clustering that takes edge weights into account can be applied. Suitable for this is *Modularity Clustering*, applied to the undirected citation network [4]. The implementation provided by the *igraph* library of the programming language R was used[6]. For this algorithm the number of expected clusters is not a required parameter. The cluster containing the seed vertex represents the set of papers given as reading recommendations. This approach will be called *MC*.

4 Analysis and results

In order to evaluate the recommendations produced by the different algorithms, an online survey was conducted. The different algorithms generated reading recommendations based on a single citation network. These recommendations were then rated by users.

4.1 Survey Setup

As a seed for the citation network a survey paper by Fortunato on graph clustering was taken. First of all, survey papers are a good starting point for scientists

[6] `http://www.inside-r.org/packages/cran/igraph/docs/fastgreedy.community`, as seen on March 24th 2015

to embark on a new field. Secondly, they usually cite many papers and thus the resulting citation networks generated from them as starting points will contain many papers. The citation network generated from this paper used a depth of three and contained 1451 vertices.

Table 1 shows the number of reading recommendations (excluding the seed paper if it was given as a recommendation) given by the different algorithms. Those marked with a † were evaluated in the online survey. All others were omitted, either because they gave too many reading recommendations – e.g. MC – or too few – e.g. LC Seed SPC*topSim.

Figure 4 shows how the seed paper and the 40 papers recommended by the algorithms MP, WMP, LC Seed and LC Intersection SPC*topSim are connected in the citation network. The papers recommended by the MP algorithm clearly form a path in the citation network – as was expected – that only diversifies in the second last node on the path. The recommendations generated by the WMP show a single path that partially overlaps with that generated by the MP algorithm. The papers recommended using LC Seed are overall not very well connected among another, but mostly only connect via the seed paper. The recommendations generated by LC Intersection SPC*topSim on the other hand overlap with all recommendations from all other three algorithms. They are not as clearly connected as those recommended by the MP or WMP, but more connected than those generated with LC Seed. Papers from different recommendations are sometimes connected, especially the papers recommended by LC Seed and LC Intersection SPC*topSim.

4.2 Evaluation

In the conducted online survey the quality of the 40 papers recommended by the four different algorithms was evaluated. At the beginning of the survey the user was provided with the author information, title and abstract, as well as reference of the seed paper by Fortunato[7]. This information, as well as all other used in the survey, was reproduced from CiteSeerX or comparable web services and was hence publicly available.

[7] Fortunato, S.: Community detection in graphs. Physics Reports 786(3), 75-174 (2010)

Algorithm	No. of Recommendations
LC Seed†	20
LC Intersection	389
LC Seed SPC*topSim	1
LC Intersection SPC*topSim†	11
MP†	14
WMP†	5
MC	217

Table 1. Number of Reading Recommendations per Algorithm

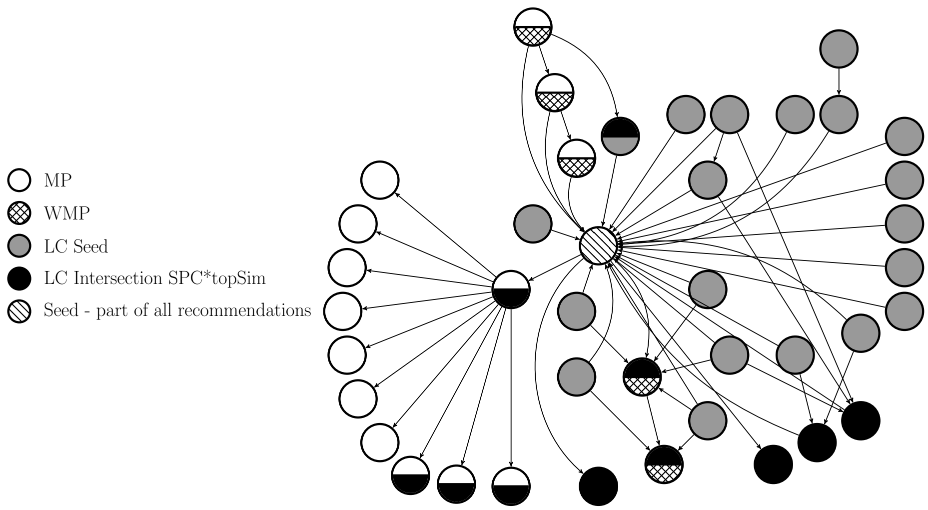

Fig. 4. The 40 papers recommended by the four algorithms evaluated in the survey and the seed node with their citations among each other

The task was that the user was to assume that he or she wanted to find further literature on the topic covered by the seed paper. He or she was then asked to provide additional information with respect to his level of expertise in computer science, graph clustering and social network analysis on a 4-point Likert scale from one to four ("No expertise at all", "I have heard of it", "I am familiar with the field" and "I am a scientific researcher in this field"). This enabled us to ensure a high level of expertise of the raters.

The survey provided the same information as for the seed paper (title, author, abstract and reference) for all 40 papers to be rated. We limited the given information to the abstracts and titles instead of providing full papers because we assumed that scientific researchers familiar with a field can estimate the topic of a paper given only the abstract and title. For each of these 40 papers the user was asked to rate the quality of each as a reading recommendation compared to the topic of the seed paper on a 5-point Likert scale from one to five ("Very Bad", "Bad", "Neutral", "Good" and "Very Good"). The information on the recommended papers alternated with the questions to rate the quality. During the whole time the information on the seed paper were visible for comparison. The user was unaware that the recommendations had been generated by different algorithms. Moreover, the papers were ordered by the last name of the first authors.

In the survey 17 raters participated with a high average expertise in computer science (3.76), a medium average expertise in graph clustering (2.76) and a medium average expertise in social network analysis (2.65). The survey evaluated the recommendation quality of the different recommended papers. The second

[8] Expertise in Graph Clustering 3 or 4 on 4-point Likert scale; 10 Raters

Algorithm	All Raters		Clustering Med-Exp[8]	
	Avg. Rating	σ	Avg. Rating	σ
MP	2.32	0.45	2.19	0.34
WMP	3.11	1.01	3.04	1.00
LC Intersection SPC*topSim	3.37	0.75	3.22	0.91
LC Seed	3.88	0.55	3.73	0.54

Table 2. Results of the first survey

	All Raters	Clustering Med-Exp[8]
Consistency	0.546	0.505
Absolute Agreement	0.507	0.482

Table 3. Interrater Reliability for Survey 1 using ICC(2,1)

and third columns of Table 2 give the standard deviations and rounded average rating results of the four algorithms obtained by averaging the ratings the papers that were recommended by a specific algorithm got. If all participants stating their level of expertise in graph clustering to be *1* or *2* are omitted – which leaves 10 participants – the average results look like given in column four of the same table. As can be seen, the results do not differ much. The interrater reliability as given in Table 3 was measured by ICC(2,1). According to Andresen [1], an ICC value between 0.4 and 0.75 is a moderate to good reliability which is the case here. However, since the number of participants was very small, a higher interrater reliability was not expected. The survey task was also formulated openly in regards to the rating of the quality of recommendations. It highly depended on the criteria used for the rating by the different raters, e.g. whether they wanted to find other papers on graph clustering in general or papers on very specialized forms of graph clustering.

Finally, the four algorithms are compared to an algorithm that generates reading recommendations while considering all papers in the citation network as candidates. A variant of the *Item-Item Collaborative Filtering* algorithm proposed by McNee et al. [12] is used to recommend papers based only on the structure of the citation network. This algorithm is used as a baseline against which to compare all other algorithms. For the citation network the (undirected) adjacency matrix is calculated. In order to recommend papers every column of the matrix is compared to the column representing the seed paper and the cosine similarity is calculated for every pair. The N most similar papers are taken as the reading recommendations. This algorithm will be referred to as *CF*.

A weighted variant – weighted CF (WCF) – was also considered, that incorporates the topic-structure similarity by encoding it in the adjacency matrix. The reading recommendations are generated as in the CF approach. For the evaluation the value of N was set to ten for both variants.

Both variants were tested in a survey structured identically to the first one. Moreover, the same dataset was used. Both variants recommended a total of

Algorithm	Avg. Rating	σ
CF	3.73	0.73
WCF	3.7	0.91

Table 4. Results of the second survey

	All Raters
Consistency	0.504
Absolute Agreement	0.471

Table 5. Interrater Reliability for Survey 2 using ICC(2,1)

14 different papers. The second study was conducted by six participants with a high average expertise of 3.83 in computer science, medium expertise in graph clustering (2.33) and a medium expertise in Social Network Analysis (2.33). Table 4 gives the rounded average rating results and standard deviations of the two algorithms. The interrater reliability as given in Table 5 was measured by ICC(2,1) and is again moderate but slightly worse than for the first survey. However, since the second survey had only six participants, a low interrater reliability was to be expected.

5 Discussion

The papers recommended by the MP algorithm received on average a low rating of (2.32, $\sigma = 0.45$) that does not qualify the results as good. Slightly better rated were the papers recommended by WMP, which were rated neutrally. This indicates that the recommendations benefited from the used topic-structure similarity, as was expected. LC Intersection SPC*topSim received a neutral-positive rating of (3.37, $\sigma = 0.75$). Yet LC Seed received the highest ratings with (3.88, $\sigma = 0.55$) or (3.73, $\sigma = 0.54$) by the experts on a 5-point Likert scale, which is considered to be a *good* rating. Moreover, the standard deviation is low. Considering, that the ratings reflect precision, the ratings of LC Seed are even better since it is the algorithm that recommended the most papers. The recall of the generated recommendations could not be measured since ground truth on the quality of all 1451 papers in the citation network was not available.

Overall, it is indicated that the LC algorithm is better suited for recommending papers than the MP algorithm. Moreover, the usage of topic-structure similarities improves the ratings, whereas the usage of Main Path's SPC weights seems to lower them.

Since the MC algorithm recommended far too many papers, it could not be evaluated in a user study. However, 13 of the 20 papers recommended by LC Seed were also recommended by the MC algorithm. One paper recommended by LC Intersection SPC*topSim was also recommended by MC. There was no overlap with the recommendations generated by MP or WMP. The small overlap with LC Intersection SPC*topSim might be explained by the usage of the different edge weights. The same holds for the nonexistent overlap with the recommendations by MP and WMP, however these two algorithms additionally only greedily select the locally highest edge weights which might also explain the difference in recommendations.

The seven recommendations not in the intersection of MC and LC Seed received an overall average rating of (3.51, $\sigma = 0.63$) if all raters are considered or (3.26, $\sigma = 0.53$) if only modularity clustering experts are considered. Hence, good and moderately good recommendations are not in the intersection. The 13 nodes in the intersection however received a rating of (4.08, $\sigma = 0.40$) or (3.98, $\sigma = 0.35$) by the experts. This is another improvement in the rating compared to the rating of all 20 papers recommended by LC Seed which is (3.88, $\sigma = 0.55$) or (3.73, $\sigma = 0.54$). Hence, focusing the results of the MC algorithm by intersecting them with the results of the LC Seed algorithm might result in a very good reading recommendation algorithm.

The survey rating the recommendation quality of the CF and WCF algorithms shows that the LC Seed algorithm was rated more or less equally compared to the CF and WCF approaches. However, since the LC Seed recommended twice as many papers, the high rating can be valued even more highly.

So far the recommendations generated by the algorithms have only been analyzed for one input paper – a survey paper from graph theory. It would be interesting to see how the type of the seed paper – survey paper, short paper etc. – and accordingly the shape of the extracted citation network affect the quality of results.

6 Conclusion and Future Work

In this paper a novel algorithm – LC – was presented to recommend scientific papers to read. It incorporates a topic-structure similarity and makes high usage of the structure of citation networks in order to recommend papers. Furthermore, two other algorithms were adapted to recommend papers based on the structure of the citation network – Modularity Clustering and Main Path Analysis. In an online user study the algorithms were evaluated. It was found that the novel algorithm produced the best results. Incorporating the SPC weights from Main Path Analysis on the other hand seemed to worsen the quality of recommendations. All algorithms were furthermore compared to an existing CF based approach to recommend papers. It was shown that the novel algorithm produced an equally high precision but generated more recommendations. The results indicate that a combination of the novel algorithm with Modularity Clustering might further improve the results.

However, since only a small amount of users participated in the studies, these results need to be verified in future surveys. In these the participants will be asked to select the seed paper themselves in order to ensure their expertise when rating the quality of the recommendations.

References

1. Andresen, E.M.: Criteria for assessing the tools of disability outcomes research. Archives of physical medicine and rehabilitation 81, 15–20 (2000)

2. Batagelj, V.: Efficient algorithms for citation network analysis. arXiv preprint cs/0309023 (2003)
3. Charles, C.: Analysis of Communication Flow in Online Chats. Master's thesis, University of Duisburg-Essen (2013)
4. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Physical review E 70(6), 066111 (2004)
5. Ding, Y., Zhang, G., Chambers, T., Song, M., Wang, X., Zhai, C.: Content-based citation analysis: The next generation of citation analysis. Journal of the Association for Information Science and Technology 65(9), 1820–1833 (2014)
6. Garfield, E.: Abcs of cluster mapping. 1. most active fields in the life sciences in 1978. Current Contents (40), 5–12 (1980)
7. Garfield, E., et al.: Can citation indexing be automated. In: Statistical association methods for mechanized documentation, symposium proceedings. pp. 189–192 (1965)
8. He, X., Zha, H., Ding, C.H., Simon, H.D.: Web document clustering using hyperlink structures. Computational Statistics & Data Analysis 41(1), 19–45 (2002)
9. Hummon, N.P., Doreian, P.: Connectivity in a citation network: The development of dna theory. Social Networks 11(1), 39–63 (1989)
10. Lee, J., Lee, K., Kim, J.G.: Personalized academic research paper recommendation system. arXiv preprint arXiv:1304.5457 (2013)
11. Liu, J.S., Lu, L.Y.: 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(3), 528–542 (2012)
12. McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proceedings of the 2002 ACM conference on Computer supported cooperative work. pp. 116–125. ACM (2002)
13. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of EMNLP. vol. 4, p. 275. Barcelona, Spain (2004)
14. Pohl, S., Radlinski, F., Joachims, T.: Recommending related papers based on digital library access records. CoRR abs/0704.2902 (2007)
15. Small, H.: Co-citation in the scientific literature: A new measure of the relationship between two documents. Journal of the American Society for information Science 24(4), 265–269 (1973)
16. Strohman, T., Croft, W.B., Jensen, D.: Recommending citations for academic papers. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 705–706. ACM (2007)
17. Sugiyama, K., Kan, M.Y.: Scholarly paper recommendation via user's recent research interests. In: Proceedings of the 10th annual joint conference on Digital libraries. pp. 29–38. ACM (2010)
18. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic classification of citation function. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. pp. 103–110. Association for Computational Linguistics (2006)
19. Torres, R., McNee, S.M., Abel, M., Konstan, J.A., Riedl, J.: Enhancing digital libraries with techlens+. In: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries. pp. 228–236. ACM (2004)
20. Wise, M.J.: Yap3: Improved detection of similarities in computer program and other texts. In: ACM SIGCSE Bulletin. vol. 28, pp. 130–134. ACM (1996)