CollabSeer: A Search Engine for Collaboration Discovery

Hung-Hsuan Chen[†], Liang Gou[‡], Xiaolong (Luke) Zhang[‡], C. Lee Giles[†‡]
[†]Computer Science and Engineering
[‡]Information Sciences and Technology
The Pennsylvania State University
hhchen@psu.edu, {lug129, lzhang, giles}@ist.psu.edu

ABSTRACT

Collaborative research has been increasingly popular and important in academic circles. However, there is no open platform available for scholars or scientists to effectively discover potential collaborators. This paper discusses CollabSeer, an open system to recommend potential research collaborators for scholars and scientists. CollabSeer discovers collaborators based on the structure of the coauthor network and a user's research interests. Currently, three different network structure analysis methods that use vertex similarity are supported in CollabSeer: Jaccard similarity, cosine similarity, and our relation strength similarity measure. Users can also request a recommendation by selecting a topic of interest. The topic of interest list is determined by CollabSeer's lexical analysis module, which analyzes the key phrases of previous publications. The CollabSeer system is highly modularized making it easy to add or replace the network analysis module or users' topic of interest analysis module. CollabSeer integrates the results of the two modules to recommend collaborators to users. Initial experimental results over the a subset of the CiteSeerX database shows that CollabSeer can efficiently discover prospective collaborators.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*relevance feedbacks, retrieval models, selection process*; H.3.7 [**Information Storage and Retrieval**]: Digital Library—*Collections, Dissemination*; J.4 [**Social and Behavior Sciences**]: Sociology

General Terms

Design, Algorithms, Experimentation

Keywords

Social Network, Coauthor Network, Graph Theory, Link Analysis, Digital Library, Information Retrieval

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
JCDL'11, June 13–17, 2011, Ottawa, Ontario, Canada.
Copyright 2011 ACM 978-1-4503-0744-4/11/06 ...$10.00.

1. INTRODUCTION

Collaboration among scholars seems to be increasing in popularity [15]. Research collaboration obviously brings more points of view to research issues addressed. More importantly, studies show that scholars with higher levels of collaboration tend to be more productive [17, 23]. Therefore, it would seem beneficial for researchers, especially young researchers, to find potential successful collaborators.

However, the design of traditional digital libraries and search engines focuses on discovering relevant documents. This design makes it not straightforward to search for people who share similar research interests. Recently, a few digital library platforms, such as Microsoft Academic Search[1] and ArnetMiner[2], return a list of experts for a particular domain. These lists, however, provide only a limited set of names and ignore the social network of experts given a particular author. To help in efficiently discovering potential collaborators, we propose a new system that considers social network structure, reachability, and research interests of users to recommend potential collaborators.

In this paper, we introduce CollabSeer[3], a search engine for discovering potential collaborators for a given author or researcher. CollabSeer is based on CiteSeerX[4] dataset to build the coauthor network, which includes over $1,300,000$ computer science related literature and over million unique authors. CollabSeer discovers potential collaborators by analyzing the structure of a coauthor network and the user's research interests. Currently, CollabSeer supports three different network structure analysis modules for collaborator search: Jaccard similarity, cosine similarity, and our relation strength similarity. Users could further refine the recommendation results by clicking on their topics of interest, which are generated by extracting the key phrases of previous publications. The system is highly modularized; thus it is easy to add or update the network structure analysis module or the topic of interest analysis module. To see the effectiveness of the system, we selected 20 information retrieval and machine learning related venues for experiments. The experimental results show that CollabSeer can suggest collaborators whose research interests are closely related to the given user by using only the network structure analysis module.

The rest of the paper is organized as follows. In Section 2, we review previous work related to complex networks and in-

[1] http://academic.research.microsoft.com/
[2] http://www.arnetminer.org/
[3] http://proc5.ist.psu.edu:8080/collabseer/
[4] http://citeseerx.ist.psu.edu/

troduce network structure based vertex similarity algorithms that will be used for CollabSeer system. The details of system infrastructure, implementation, and user interface of CollabSeer are given in Section 3. Sections 4 explains the relation strength measure, lexical similarity measure, and how we integrate the two similarity measures. Experimental results in Section 5 evaluate the relationship between vertex similarity and lexical similarity in order to determine the effectiveness of vertex similarity measures. Summary and future work appears in Section 6.

2. RELATED WORK

Because of their importance in the CollabSeer system, we review previous work related to complex network analysis and network structure based vertex similarity measures.

2.1 Complex Network Analysis

Complex networks have been studied and utilized in several areas, such as social networks [20, 24], the world wide web [3], biological networks [4], and coauthor networks [15]. It has been shown that networks in real world scenarios have distinctive statistical and structure characteristics, such as power law distributions [2], small world phenomenon [35], community structure [10], and spatial models [9]. Recently, the evolution of network topology has been explored [8, 15, 20]. For a survey, please see [1, 5, 26].

Complex network measures can be used for coauthor network analysis. For example, the degree centrality, betweenness centrality, and closeness centrality to indicate the importance of an author [28] have been used. The similarity, difference and evolution of the statistics of coauthor network in various domains and in various digital libraries have been compared [15, 25, 27].

2.2 Vertex Similarity Analysis

Vertex similarity defines the similarity of two vertices based on the structure of network. It has been used in several areas, such as social network analysis [21], information retrieval in world wide web [19], and collaborative filtering [31]. One measure is the (normalized) number of common neighbors [29, 30, 33]. Although these methods consider only local information, they are computationally efficient. As example is the well known Jaccard similarity [33] defined in Equation 1.

$$S_{Jaccard}(v_i, v_j) = \frac{\Gamma(m_i \cap m_j)}{\Gamma(m_i \cup m_j)}, \quad (1)$$

where m_i is the set of neighbors of vertex v_i and m_j is the set of neighbors of vertex v_j. The $\Gamma()$ function returns the number of elements in the set. Jaccard similarity is based on the intuition that two vertices are more similar if they share more common neighbors. Another similarity measure, cosine similarity [30], is based on the same idea. Cosine similarity is defined as follows.

$$S_{cosine}(v_i, v_j) = \frac{\Gamma(m_i \cap m_j)}{\sqrt{\Gamma(m_i)\Gamma(m_j)}}. \quad (2)$$

Previous studies show that cosine similarity generally performs better than Jaccard similarity in most practical situations [13]. Topology overlap similarity also uses neighborhood information, such as in metabolic networks [29].

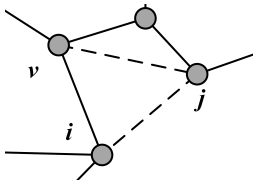


Figure 1: Diagram of vertex similarity

Topology overlap similarity is defined in Equation 3.

$$S_{t.o.}(v_i, v_j) = \frac{\Gamma(m_i \cap m_j)}{\min\left(\Gamma(m_i), \Gamma(m_j)\right)}. \quad (3)$$

Most of the local information based similarity measures exploit a similar idea [39]. Instead of using local neighborhood information, the global network structure can be used for vertex similarity calculation. We introduce three global structure based vertex similarity measures, SimRank [16] Leicht-Holme-Newman (LHN) vertex similarity [21], and P-Rank [38]. These three methods consider the vertex similarity measure based on the same intuition: two vertices are similar if their immediate neighbors in the network are themselves similar. Specifically, as shown in Figure 1, vertex i and v are connected (solid line), but v and j, i and j are not (dashed line). Then, how i is similar to j is dependent on how v, the neighbor of i, is similar to j. As such, the calculation of SimRank, LHN, and P-Rank are all recursive process, because the similarity between vertex v and j is related to the similarity between all the neighbors of v and j. While the intuition is the same, SimRank only calculates vertex similarity between vertices with even path length [21] and only makes use of in-link relationships [38], which could make a substantial difference for the final similarity score. Although global structure based similarity measures could get a larger picture regarding the entire network, the required time complexity is prohibitive. Moreover, a small structure change, such as adding a new vertex or a new edge, will eventually propagate the effect to the whole network. Therefore, it is not feasible to apply these algorithms to a large scale dynamic network. Recent work proposed a non-iterative approximation for SimRank [22] using incremental updating. However, this method allows only link updating, i.e., it assumes the total number of nodes in graph is fixed. Other work [11, 12] approximates LHN similarity by clustering the social network into virtual nodes to reduce the graph size.

Recently, network structure information was used to infer the missing links in coauthor networks [37, 39]. However, lexical similarity between authors was not considered. A recent paper [32] utilized the frequency of key phrases in author's pervious publications to infer his or her research interests. This work recommends papers, not potential collaborators, for a given user. He et. al [14] proposed a context-aware citation recommendation prototype. Their model suggests papers based on content, not coauthor network or citation network information. Cucchiarelli and D'Antonio [6] utilized the coauthor links and similarity links formed by the centroids of the documents to discover partnership opportunities among research units. To the best of our knowledge, CollabSeer is the first online system for discovering prospective collaborators for individuals.

3. SYSTEM OVERVIEW

CollabSeer is built based on CiteSeerX dataset. To minimize the impact of author name ambiguity problem, the random forest learning [34] is used to disambiguate the author names so that each vertex represents a distinct author.

3.1 System Architecture

Figure 2 shows the system architecture of CollabSeer. For the user interface, users put in queries and receive collaboration recommendations from CollabSeer system. The CollabSeer system consists of the following five components:

Coauthor Information Analyzer retrieves data from CiteSeerX dataset to build a weighted coauthor network, where each node acts as an author, each edge represents a collaboration behavior, and the weight of an edge indicates the number of coauthored articles of the two authors.

Vertex Similarity analyzes the structure of the coauthor network and indexes the result for later use. Currently, CollabSeer supports three vertex similarity modules; they are Jaccard similarity [33], cosine similarity [30], and our relation strength similarity. Other similarity measures [16, 21] can also be added as modules. We introduce the details of relation strength similarity in Section 4.1. Though targeted at coauthor networks, our proposed vertex similarity measure can be applied to other complex network applications as well.

Key Phrase Extractor analyzes the scientific literatures to get the key phrases of each article by KEA [36] algorithm.

Lexical Similarity associates the authors to the key phrases for lexical similarity analysis. The result is indexed to handle the real time queries. We introduce the details in Section 4.2.

Similarity Integrator amalgamates the indexed vertex similarity score and the indexed lexical similarity score to calculate the collaboration recommendation in real time.

3.2 User Interface

Here we introduce the user interface and design considerations. Figure 3 shows the screenshots of the system. Figure 3(a) is first page the users would see when they visit CollabSeer. Users could put names in the input box, and select one of the vertex similarity measures shown in the drop down list. The CollabSeer system would suggest the potential collaborators based on user's selected measure. Figure 3(b) shows a list of matched names. Note that different authors may share the same name either as full names or as initials and last names. We disambiguate the author names using random forest [34]. The snapshot of the recommendation list and user's topic of interest are displayed in Figure 3(c). The lower part of Figure 3(c) shows the list of recommended collaborators with their service institutions. The upper part is the user's topic of interest arranged in an alphabetical order. The size of the key phrases are proportional to the significance of interests, a metric to measure strength of user's interest on this topic. Details about significance of interests calculation will be introduced in Section 4.2. When a user clicks on any of the topics, CollabSeer reranks the recommendation list based on both vertex similarity score and lexical similarity score. A user could also click on the names of the potential collaborators to see more information, including how the two users are related, as illustrated in Figure 3(d).

4. SIMILARITY ALGORITHMS

4.1 Vertex Similarity Algorithm

4.1.1 Relation Strength Vertex Similarity

The relation strength similarity is based on the idea of *relation strength*, which defines how close two adjacent vertices are. For the coauthor network in particular, two adjacent vertices indicate two people coauthored at least one article together before. The relation strength of two adjacent authors is proportional to the number of their coauthored articles. Assuming user A has n_A publications, user B has n_B publications, user A and user B coauthored n_{AB} articles. The relation strength from author A to author B is defined as follows.

$$R(A,B) := \frac{n_{AB}}{n_A}. \tag{4}$$

For two non-adjacent authors A and C, if A could reach C only through author B, then how close author A to author C should be proportional to the relation strength of author A to author B and the relation strength of author B to author C. We define *indirect relation strength* from author A to author C as

$$R^*(A,C) := R(A,B)\cdot R(B,C) = \frac{n_{AB}}{n_A}\cdot\frac{n_{BC}}{n_B}. \tag{5}$$

Equation 5 can be generalized as follows. Assuming there exists a simple path p_m from A to C, where the path p_m is formed by A, B_1, B_2, $\cdots$, B_K, C. The indirect relation strength from A to C through simple path p_m is

$$R^*_{p_m}(A,C) := R(A,B_1)\cdot\prod_{k=1}^{K-1} R(B_k,B_{k+1})\cdot R(B_K,C). \tag{6}$$

Now, if there are M distinct simple paths $p_1, p_2, \ldots, p_M$ from A to C , the similarity value from A to C is defined as

$$S(A,C) := \sum_{m=1}^{M} R^*_{p_m}(A,C). \tag{7}$$

4.1.2 Analysis of Relation Strength Similarity

First, we now show that the similarity measure between any two nodes is always between 0 and 1. For any neighboring pair, the relation strength is not larger than 1 by Equation 4. Assuming that there is at least one path p_m between two vertices v_i and v_j. The indirect relation strength of v_i to v_j through p_m is still not larger than one because it is defined as the products of relation strength by Equation 6. We can rewrite Equation 7 as follows to show that $S(v_i,v_j)$ is always less or equal 1.

$$\begin{aligned} & S(v_i,v_j) = \textstyle\sum_{m=1}^{M} R^*_{p_m}(v_i,v_j) \\ = \;& \textstyle\sum_{m=1}^{M}\left[R(v_i,v_{i+1}^{(m)})\prod_{k=1}^{K-1} R(v_{i+k}^{(m)},v_{i+k+1}^{(m)})R(v_K^{(m)},v_j)\right] \\ \leq \;& \textstyle\sum_{m=1}^{M} R(v_i,v_{i+1}^{(m)}) \\ \leq \;& 1, \end{aligned} \tag{8}$$

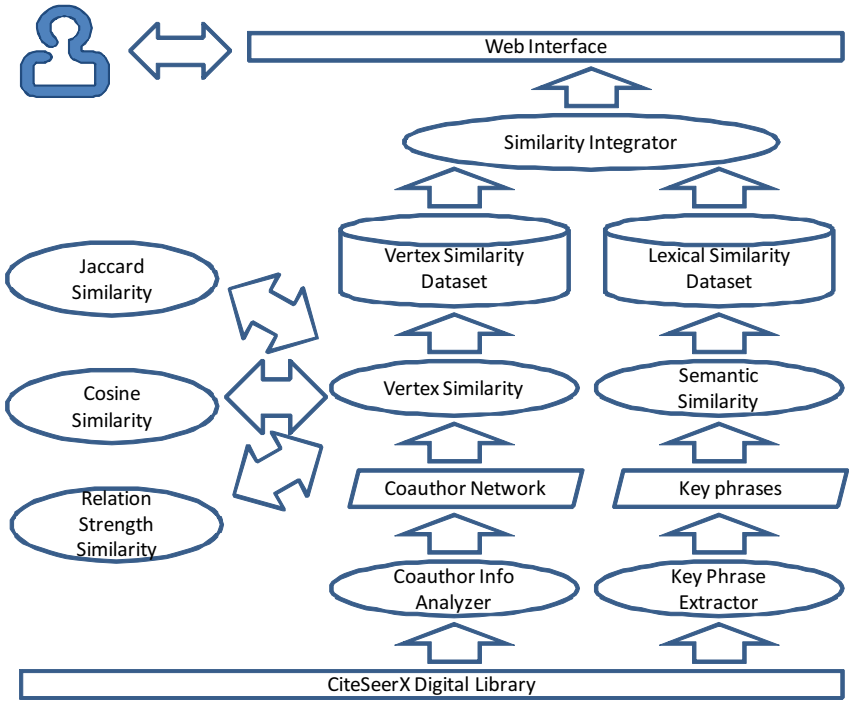


Figure 2: CollabSeer system framework

where $v_i, v_1^{(m)}, v_2^{(m)}, \ldots, v_K^{(m)}, v_j$ form p_m, the mth path between vertex v_i and vertex v_j.

Our vertex similarity measure considers both relation strength and reachability between two vertices. The relation strength is included in the measure by Equation 4, where the more papers two people coauthored, the stronger the relation strength is. The reachability between two vertices is included by Equation 6, where the longer the path, the smaller the indirect relation strength tends to be. This is because indirect relation strength is defined as the product of relation strength, which is between 0 and 1.

Compared with other similarity measures, such as Jaccard similarity [33], cosine similarity [30], SimRank [16], LHN [21], or P-Rank [38], relation strength similarity has at least the following two advantages. First, It is asymmetric, i.e., $S(A, B)$ may not equal $S(B, A)$. This is because the relation strength from author A to author B may not equal the relation strength from B to A, as defined in Equation 4. The asymmetric property is closer to the real world scenario. For instance, suppose author A only cooperates with author B; author B works with author A, author C, and author D. Since author A has only one choice and B has several options, the importance of author B to author A is larger than the importance of author A to author B. Second, relation strength similarity considers the edges' weights, which can be used to represent the number of coauthored papers between two authors. Suppose that the number of papers coauthored between author A and author B is larger than the coauthored papers between author B and author C, author A should be more important to author B. Previous works [16, 21, 30, 33] would regard A and C be equally important because they can only deal with unweighted graph and hence ignore the number of coauthored papers.

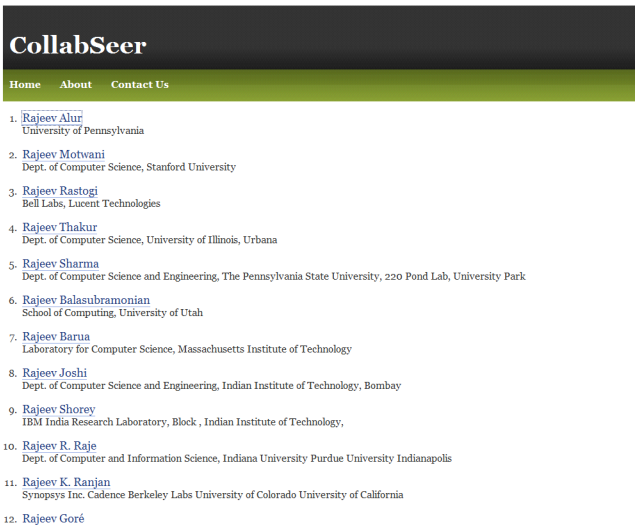
Although this similarity measure takes into consideration the complete topology of the network, the complexity to compute the similarity from one vertex to all the other vertices is $O(d^\ell)$, where d is the average degree of vertices and ℓ is the longest path length between two vertices. This large complexity comes from the need to retrieve all the available simple paths between two vertices. Since the similarity measure is asymmetric, we need to compute the similarity vector for all the nodes. Therefore, we need $O(nd^\ell)$ to compute the similarity between all the nodes in the graph.

The formidable parts of the time complexity comes from the exponent ℓ, the longest path length between two vertices. We approximate the relation strength similarity by introducing a new *discovery range* parameter, r, to control the maximum degree of separation for collaborators, i.e., we only look for collaborators at most r hops away. The required time complexity becomes $O(nd^r) \approxeq O(n)$ when $d << n$ and $r << n$. The approximation is reasonable because once the path length is too long, the product form in Equation 6 would make $R^*_{p_m}$ very small, and therefore contributes little to the final similarity measure (Equation 7). In current CollabSeer system, we set the value to 3, i.e., our approximation looks for nodes in three degrees of separation. Compared with previous work, the local information based vertex similarity algorithms [13, 30, 33] are too restrictive in the sense that they only look for authors who share mutual friends with the given author and fail to consider the global picture of the network. However, the global information based vertex similarity algorithms [16, 21, 38] are not computationally feasible for large networks. Our algorithm allows users to control the discovery range and thus reduces the complexity.

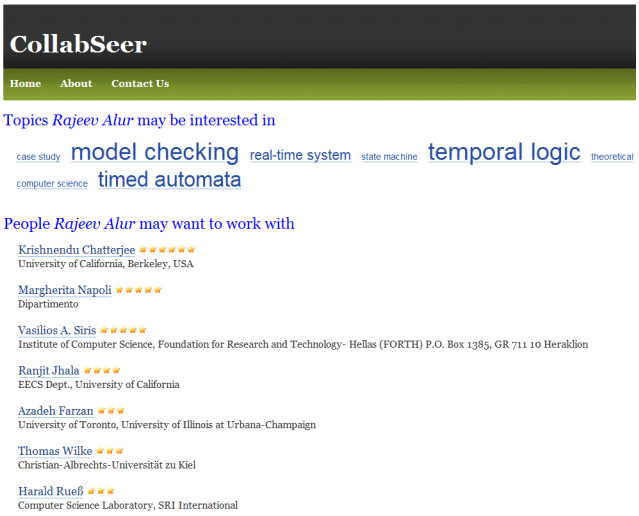
4.2 Lexical Similarity Algorithm

In addition to using vertex similarity to find potential col-

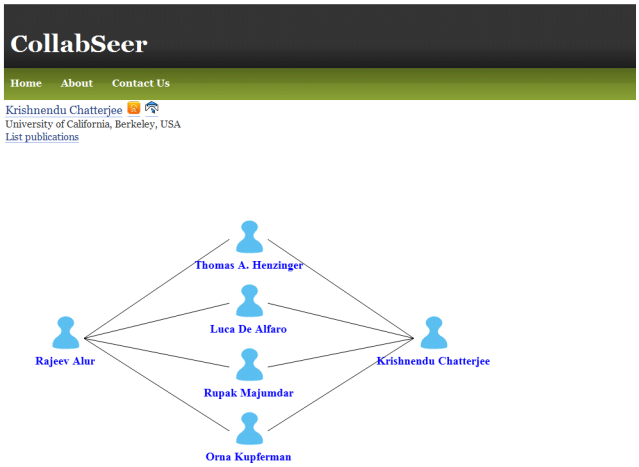
(a) The main query interface. Different vertex similarity modules are available in the drop down list.



(b) A list of matched names



(c) A list of recommended collaborators and user's topic of interests



(d) The relationship between two users

Figure 3: Snapshots of CollabSeer system

laborators, CollabSeer also allows users to select topics of interests in order to refine the recommendations.

First, CollabSeer extracts the key phrases of each document [36], and associates the authors of the document to the key phrases. On the one hand, CollabSeer can use the table to infer an author's research interests. The more frequently a key phrase associates with an author, the more likely the author would be interested in the topic. On the other hand, CollabSeer can gauge an author's contribution to a specific topic. The contribution score is calculated using the number of times the key phrase associating with the author divided by the total number of times the key phrase appears.

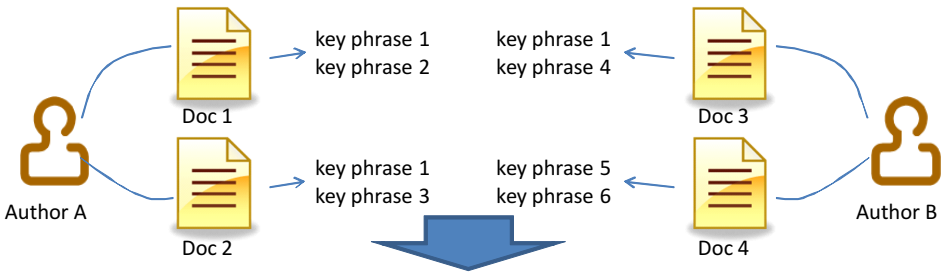
Specifically, assuming that author A_i has published m papers $p_1, p_2, \ldots, p_m$. The key phrases extracted from the m papers are $k_1, k_2, \ldots, k_n$, with the number of appearing times be $f_1, f_2, \ldots, f_n$ respectively. CollabSeer determines author A_i be interested in topics be $k_1, k_2, \ldots, k_n$. The *significance of interests* ($S.O.I.$) of topic k_j $(1 \leq j \leq n)$ for author A_i is defined as

$$S.O.I(A_i, k_j) := \frac{f_j}{\sum_{k=1}^{n} f_k}. \quad (9)$$

On the other hand, suppose that topic k_j interests u authors, namely $A_1, A_2, \ldots, A_u$. For a particular author A_v $(1 \leq v \leq u)$, g_v out of author A_v's publications has key phrase k_j. CollabSeer gauges the *contribution of topic* ($C.O.T.$) for author A_v to topic k_j be

$$C.O.T.(A_v, k_j) := \frac{g_v}{\sum_{w=1}^{u} g_w}. \quad (10)$$

Let us take a closer look at the author to key phrase mapping table with an example shown in Figure 4. Assuming author A has two publications Doc 1 and Doc 2. Doc 1 has two key phrases, "key phrase 1" and "key phrase 2", and Doc 2 has two key phrases, "key phrase 1" and "key phrase 3". Author B also has two publications, Doc 3 and Doc 4. Doc 3 has key phrases "key phrase 1" and "key phrase 4", and



People	Key phrase	Frequency
Author A	Key phrase 1	2
Author A	Key phrase 2	1
Author A	Key phrase 3	1
Author B	Key phrase 1	1
Author B	Key phrase 4	1
Author B	Key phrase 5	1
Author B	Key phrase 6	1

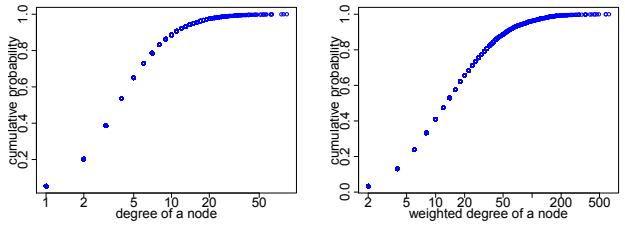
Figure 4: Illustration of how the author to key phrase map is generated

Doc 4 has key phrases "key phrase 5" and "key phrase 6". CollabSeer uses these information to build a table of people to key phrases to frequency. For instance, "key phrase 1" appears in both of author A's publications, therefore author A would be associated to "key phrase 1" with frequency 2. The rest of the tables are shown in Figure 4. In the example, CollabSeer infers author A's research interests are topics related to "key phrase 1", "key phrase 2", and "key phrase 3", with significance of interests be 2/4, 1/4, and 1/4 respectively. Author B's research interests are topics related to "key phrase 1", "key phrase 4", "key phrase 5", and "key phrase 6", with significance of interests be 1/4 for all four topics. For the topic related to "key phrase 1", author A contributes twice as much as author B, since 2/3 of the papers related to "key phrase 1" is published by author A and 1/3 of them is published by author B.

4.3 Integration of Vertex Similarity and Lexical Similarity

CollabSeer considers both vertex similarity and lexical similarity to recommend the collaborators. For two authors A_i and A_j in the coauthor network, CollabSeer normalizes the vertex similarity score to be between 0 and 1. The currently supported vertex similarity scores (Jaccard similarity [33], cosine similarity [30], and relation strength similarity) are all between 0 and 1 by their nature so we could ignore the normalization step, but we may still need the step when other similarity measures are added to CollabSeer. CollabSeer system lists the recommended collaborators by only vertex similarity score in default.

CollabSeer lists the topics the user might be interested in based on his or her previous publication history using Equation 9. The value is also between 0 and 1 by its nature. When the user clicks on any of these terms, CollabSeer calculates contribution of topic (C.O.T.) for authors to a particular topic by Equation 10. The C.O.T. score $S_{C.O.T.}$ is



(a) Empirical CDF of vertices' degrees (unweighted) (b) Empirical CDF of vertices' weighted degree

Figure 5: Empirical cumulative distribution function of vertices' unweighted and weighted degrees

integrated with the vertex similarity score $S_{v.s.}$ by

$$S := \exp(S_{v.s.}) \cdot \exp(S_{C.O.T.}). \quad (11)$$

We use the product of exponential functions instead of product of the two similarity scores because we don't want the zero vertex similarity score or zero lexical similarity score to zeroize the whole measure. In addition, since $S_{v.s.}$ and $S_{C.O.T.}$ are both normalized between 0 and 1, they are equally important for the final score.

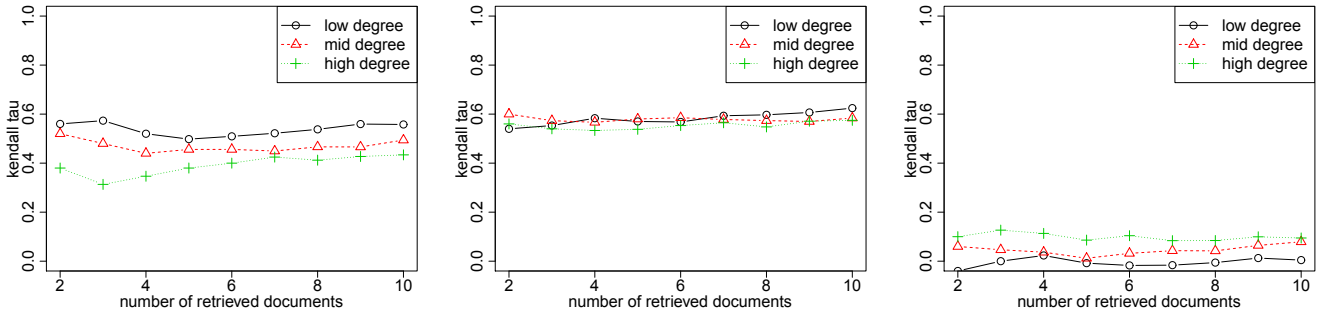
5. EXPERIMENTS

5.1 Experiment Data Collection

CollabSeer system makes use of CiteSeerX data to build the coauthor network. We use random forest learning [34] to disambiguate the author names. Our system now has a coauthor network containing more than 300,000 distinct authors. The giant component (the largest connected subgraph) accounts for 69.1% of the coauthor network. For experiments, we select 20 information retrieval, machine learning, and

Table 1: The statistical properties of the coauthor networks

Name	#Papers	#Authors	Avg Deg	Avg Cluster Coef.	Avg Path Length
CollabSeer	1,321,190	308,116	6.63	0.58	–
Giant Comp. of CollabSeer	942,308	212,881	7.94	0.64	6.62
The selected 20 venues	285,550	11,932	3.75	0.61	–
Giant Comp. of the 20 venues	146,420	5,611	5.36	0.70	7.43



(a) The result of Jaccard similarity measure (b) The result of cosine similarity measure (c) The result of relation strength similarity measure

Figure 6: The results of different similarity measures for high, mid, and low degree nodes

data mining related conferences[5] published between 1979 and 2007 to construct an coauthor network. The statistical properties of the CollabSeer, the experimental coauthor network, and their giant components are listed in Table 1. In our later experiments, we will compare the similarity for vertices with different degrees. To let the readers get more ideas about the degrees of the network vertices, we show the empirical cumulative distribution function of vertices' degrees in Figure 5. Figure 5(a) shows the empirical cumulative distribution function of the degree of the nodes; the weights of the edges are ignored. Figure 5(b) illustrates the empirical cumulative distribution function for weighted degree, i.e., the sum of edge weights adjacent to the node. Note that the x axis is in logarithm scale for better visualization purpose.

5.2 Evaluation

The difficulty of evaluating different similarity measures is because vertex similarity results usually lack interpretability [7]. One method would be a user study. Another method is to create a gold standard lexical similarity as the ground truth and comparing the consistency of vertex similarity with the lexical similarity.

To generate the gold standard lexical similarity, we build a text vector for each vertex based on the vocabularies used in the title of authors publications with stopwords are removed. We use Euclidean distance to represents the difference of two text vectors. Specifically, for two vertices v_i and v_j and their associating text vectors X_i and X_j, the Euclidean distance is

$$d(X_i, X_j) = \sqrt{\sum_{\forall k}(x_{ik} - y_{jk})^2}, \tag{12}$$

[5]The 20 conferences are: AAAI, CIKM, ECIR, EDBT, ICDE, ICDM, ICDT, IJCAI, JCDL, KDD, NIPS, PAKDD, PKDD, PODS, SDM, SIGIR, SIGMOD, UAI, VLDB, and WWW.

where $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $X_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$, the pair (x_{ik}, x_{jk}) is the appearing frequency of key phrase w_k in v_i and v_j's publication titles.

We use the distance between text vectors as an indicator about similarity of two vertices. The closer the distance, the more similar two vertices are. We use Kendall tau rank correlation coefficient [18] to compare the ranking of different vertex similarity measures. Kendall tau is a statistic used to measure the ranking correlation between two quantities. Specifically, we use Kendall tau type b because it makes adjustments for ties. For two sequences X and Y, $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$, we say two pairs (x_i, y_i) and (x_j, y_j) are concordant if both $x_i > x_j$ and $y_i > y_j$, or if both $x_i < x_j$ and $y_i < y_j$. We say the two pairs are discordant if $x_i > x_j$ and $y_i < y_j$, or $x_i < x_j$ and $y_i > y_j$. Kendall tau b is defined as

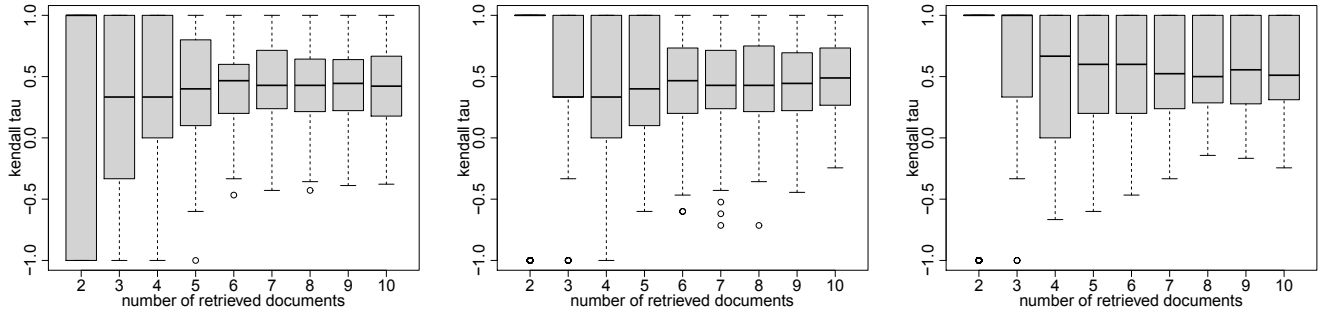
$$\tau_b(X, Y) = \frac{n_c - n_d}{\sqrt{(n_c + n_d + t_x) \cdot (n_c + n_d + t_y)}}, \tag{13}$$

where n_c is the number of concordant pairs, n_d is the number of discordant pairs, t_x is the number of pairs tied only on the first data sequence, t_y is the number of pairs tied only on the second data sequence.

The Kendall tau statistic is always between -1 and 1, where 1 means the ranking of two sequences perfectly match each other, and -1 means the ranking of one sequence is the reverse of the other.

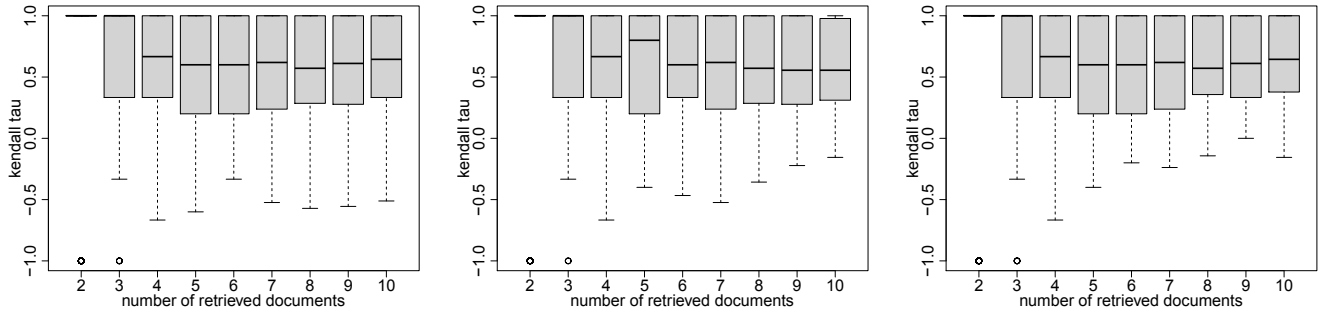
5.3 Experimental Results

Referring to Figure 5(a), we classify the vertices into high, mid, and low degree vertices. High degree vertices are those with degree numbers in the top 1/3 of all the vertices; low degree vertices are those with degree numbers in the bottom 1/3 of all the vertices; and mid degree vertices are all the remaining. We randomly pickup 100 high degree nodes, 100 mid degree nodes, and 100 low degree nodes for the following experiment.



(a) Kendall tau boxplot of high degree nodes (b) Kendall tau boxplot of mid degree nodes (c) Kendall tau boxplot of low degree nodes

Figure 7: Boxplot of Jaccard similarity for high, mid, and low degree nodes



(a) Kendall tau boxplot of high degree nodes (b) Kendall tau boxplot of mid degree nodes (c) Kendall tau boxplot of low degree nodes

Figure 8: Boxplot of cosine similarity for high, mid, and low degree nodes

Figure 6 shows the performance of similarity measures for high, mid, and low degree vertices. For each (x, y) pair in the figure, y is the Kendall tau statistic between the vertex similarity and the background truth (lexical similarity) for the first x returns. Each point is the average of 100 results. Note that the x axis starts from 2, since Kendall tau value is defined only if the sequences have more than 1 items.

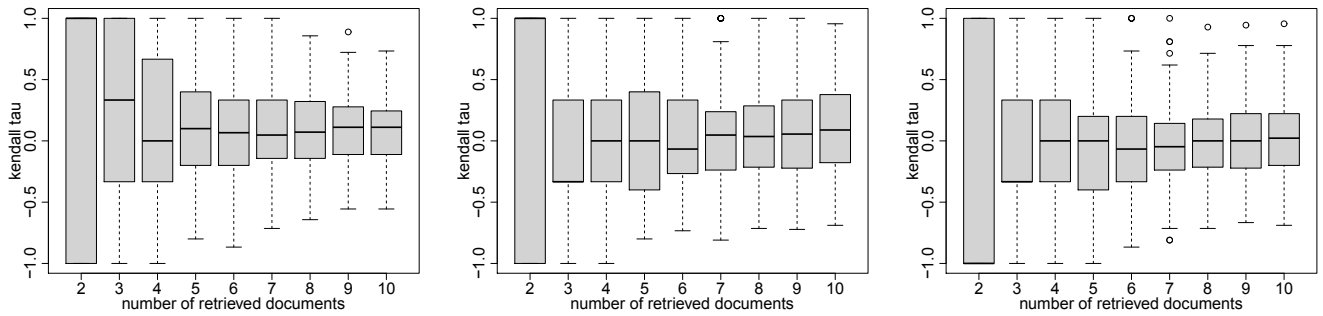
From Figure 6(a) and Figure 6(b), Jaccard similarity and cosine similarity shows similar performance in general. It is not surprising since Jaccard similarity (Equation 1) and cosine similarity (Equation 2) share the same numerator. Note that the cosine similarity measure constantly performs slightly better than Jaccard similarity for high degree and mid degree nodes. This is mainly because the variance of Kendall tau result for Jaccard similarity is larger than the result of cosine similarity, as shown in Figure 7 and Figure 8. This result matches the theoretical derivation that cosine similarity is usually more effective comparing to Jaccard similarity for link analysis in most practical cases [13].

Figure 6(c) shows the average Kendall tau result of relation strength similarity measure. The Kendall tau score of relation strength similarity is lower than Jaccard similarity and cosine similarity. It is because both Jaccad similarity and cosine similarity consider only vertices with 2 degree of separation, whereas relation strength similarity measure includes vertices with 2 degree of separation and 3 degree of separation in the experiment. Larger degree of separation makes CollabSeer system explore more potential collaborators, but also means the recommended collaborators would be more diverse in terms of their research interests. The diversity can also be suggested by Figure 9, where the variation of the Kendall tau result for relation strength similarity is generally larger than Jaccard and cosine similarity. Although the association relationship between relation strength similarity and the lexical similarity is not as strong as Jaccard or cosine coefficient, they are still mostly positively related, especially for high degree and mid degree nodes.

6. CONCLUSION AND DISCUSSION

In this paper, we introduce CollabSeer, a system that considers both the structure of a coauthor network and an author's research interests for collaborator recommendation. While expert recommendation systems usually report a similar set of users to those who share similar research interests, CollabSeer is more personalized because it suggests a different list of collaborators to different users by considering their position in the coauthoring network structure. Currently, CollabSeer supports three vertex similarity measures: Jaccard similarity, cosine similarity, and relation strength similarity. Other vertex similarity measures can also be integrated into CollabSeer easily since CollabSeer is highly modularized. The lexical similarity module utilizes the publication history to determine authors' research interests and authors' contribution to different topics.

Compared to other common vertex similarity algorithms, our relation strength similarity measure has the following advantages. It is an asymmetric similarity which allows measure to be used in more general social network applications. It can be employed on a weighted network. The relation strength between neighboring vertices can be represented



(a) Kendall tau boxplot of high degree nodes (b) Kendall tau boxplot of mid degree nodes (c) Kendall tau boxplot of low degree nodes

Figure 9: Boxplot of relation strength similarity for high, mid, and low degree nodes

by edge's weight. For coauthor network in particular, edge weights can represent the number of coauthored papers. The relation strength similarity considers reachability between any two vertices. Finally, the "discovery range" parameter can be adjusted for further collaboration exploration. For our application, increasing this parameter would recommend potential collaborators; whereas decreasing it would significantly reduce the computation.

Experimental results show that vertex similarity is positively related to lexical similarity, which means the vertex similarity measure alone could discover authors who share similar research interests. Compared with Jaccard similarity or cosine similarity, the relation strength similarity measure has a lower correlation with lexical similarity measure. This is because relation strength similarity discovers potential collaborators with larger degree of separation than Jaccard similarity or cosine similarity. This allows CollabSeer to explore more potential collaborators, but also means that the research interests of the returned authors would be more diverse. We argue that Jaccard similarity and cosine similarity are too restrictive because they only look for authors who share common friends. Since CollabSeer system allows users to choose the topic of interest to refine the recommendation, relation strength similarity permits the user to explore more authors as the candidates.

Future work could integrate CollabSeer with other vertex similarity measures, such as taking a paper's publication year into consideration for both the vertex similarity measure and lexical similarity measure. For vertex similarity, authors who collaborate recently could then have a larger relation strength than authors whose work was long in the past. For lexical similarity, authors may be more interested in topics related to their recent papers than their older work. Other lexical similarity measures are also of interest. In addition user studies can evaluate different vertex similarity measures and the design of various user interfaces.

7. ACKNOWLEDGEMENTS

We gratefully acknowledge partial support from Alcatel-Lucent and NSF.

8. REFERENCES

[1] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999.

[3] A. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77, 2000.

[4] A. Barabási and Z. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[5] S. Boccalettia, V. Latorab, Y. Morenod, M. Chavezf, and D. Hwanga. Complex networks: structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[6] A. Cucchiarelli and F. D'Antonio. Mining Potential Partnership through Opportunity Discovery in Research Networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 404–406. IEEE, 2010.

[7] C. Desrosiers and G. Karypis. Enhancing link-based similarity through the use of non-numerical labels and prior information. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 26–33. ACM, 2010.

[8] S. Dorogovtsev and J. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002.

[9] M. Gastner and M. Newman. The spatial structure of networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 49(2):247–252, 2006.

[10] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821, 2002.

[11] L. Gou, H. Chen, J. Kim, X. Zhang, and C. Giles. Sndocrank: a social network-based video search ranking framework. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pages 367–376. ACM, 2010.

[12] L. Gou, X. Zhang, H. Chen, J. Kim, and C. Giles. Social network document ranking. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, pages 313–322. ACM, 2010.

[13] L. Hamers, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, and A. Vanhoutte. Similarity measures in scientometric research: the jaccard index

versus salton's cosine formula. *Information Processing & Management*, 25(3):315–318, 1989.

[14] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 421–430. ACM, 2010.

[15] J. Huang, Z. Zhuang, J. Li, and C. Giles. Collaboration over time: characterizing and modeling network evolution. In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 107–116. ACM, 2008.

[16] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543. ACM, 2002.

[17] J. Katz and B. Martin. What is research collaboration? *Research Policy*, 26(1):1–18, 1997.

[18] M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81, 1938.

[19] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[20] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. *Link Mining: Models, Algorithms, and Applications*, pages 337–357, 2010.

[21] E. Leicht, P. Holme, and M. Newman. Vertex similarity in networks. *Physical Review E*, 73(2):26120, 2006.

[22] C. Li, J. Han, G. He, X. Jin, Y. Sun, Y. Yu, and T. Wu. Fast computation of simrank for static and dynamic information networks. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 465–476. ACM, 2010.

[23] A. Lotka and W. A. of Sciences. The frequency distribution of scientific productivity. Washington Academy of Sciences, 1926.

[24] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29–42. ACM, 2007.

[25] M. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404, 2001.

[26] M. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.

[27] M. Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5200, 2004.

[28] E. Otte and R. Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441, 2002.

[29] E. Ravasz, A. Somera, D. Mongru, Z. Oltvai, and A. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551, 2002.

[30] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. 1989.

[31] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.

[32] K. Sugiyama and M. Kan. Scholarly paper recommendation via user's recent research interests. In *Proceedings of the 10th Annual Joint Conference on Digital libraries*, pages 29–38. ACM, 2010.

[33] P. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.

[34] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 39–48. ACM, 2009.

[35] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[36] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, pages 254–255. ACM, 1999.

[37] T. Wohlfarth and R. Ichise. Semantic and event-based approach for link prediction. *Practical Aspects of Knowledge Management*, pages 50–61, 2008.

[38] P. Zhao, J. Han, and Y. Sun. P-rank: a comprehensive structural similarity measure over information networks. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 553–562. ACM, 2009.

[39] T. Zhou, L. Lü, and Y.-C. Zhang. Predicting missing links via local information. *The European Physical Journal B-Condensed Matter and Complex Systems*, 71(4):623–630, 2009.