RecAuth: Recommending Authors Based on Keywords

Ang Li
angli5@illinois.edu

Yaning Liang
liang33@illinois.edu

Abstract

Ample references should be collected prior to conducting research and writing academic papers. The references should be of high quality, which are often produced by authors who are highly reputable in their respective field of research. Less experienced researchers would benefit greatly from a tool that gives them suggestions on the authors that share their research interests. A framework, called RecAuth*, was developed to recommend a list of authors who have high relevance to the key phrases and have produced highly influential work based on an input set of keywords. The algorithm leveraged the propagation of ranking over venues and authors to generate author rankings. It was found that the recommended authors are highly consistent with the top ranking authors found on Google Scholar.*

1. Introduction

It it critical for researchers to have a good understanding of the work that has already been done in their field of research. Reading the influential work done by highly reputable authors provide a foundation for future work. However, there are many very well-known authors across all disciplines, which makes it difficult for those who are relatively inexperienced to determine which reputable authors' work they should read. A system that can automatically recommend authors based on series of keywords would be very beneficial for those who are looking for some guidance.

Current search engines such as Google Scholar are only capable of returning papers as results. Sometimes a lot of additional effort is required to filter out irrelevant papers. In addition, these engines' functionality to search papers by authors require the user to know the authors' names ahead of time. There is thus a need for systems that makes author recommendations can effectively return a small number of authors that have produced highly relevant work.

There is currently no author recommendation systems using keyword-based queries. Most of the relevant work deals with citation recommendation. For example, ClusCite is an information network-Based clustering citation system[11]. Although it is able to effectively make citation recommendations, it does not solve the aforementioned problem completely.

We developed a framework, called RecAuth, that answers keyword-based queries and recommends reputable authors who have done work closely related to the query terms. From a DBLP dataset, key words were first extracted from paper abstracts. A heterogeneous bibliographic network was then built using the extracted terms, authors, and venues for each paper. Through the network, authors were associated with venues and coauthors. Similarity scores for each author was then calculated based on the relationship between the query and the papers the authors have written. The authors were then ranked by performing iterations, where authority was propagated between authors and venues and between authors and coauthors. Upon convergence, the highest ranked authors were returned as the recommendation result.

Experiments were performed to tune various input parameters. The performance of the framework was then examined via a series of sample inputs. The same queries were also executed on Google Scholar. The authors recommended by RecAuth was compared against the authors of the papers returned by Google Scholar. It was found that authors recommended by our framework were some of the most influential and reputable authors who have produced work in fields related to the input query. These authors matched many of the authors of the papers returned by Google Scholar. In some cases, the list of authors recommended byRecAuth was more comprehensive.

2. Background and Related Work

2.1. Background Information

2.1.1 Text Mining

Some concepts related to text and phrase mining. While developing our framework, phrase mining techniques are performed to extract terms from abstracts

because when calculating similarities between papers and input keywords, it is essential to have an accurate representation of papers using a concise list of key phrases. Possible techniques include TurboTopics, KERT, ToPMine and SegPhrase+[3, 5, 4]. Having explored these techniques to perform phrase mining, SegPhrase+ had excellent performance and was incorporated RecAuth.

SegPhrease+ is an algorithm used to extract high quality phrases from a large set of input documents. Since this algorithm requires very little supervision, and the quality of result produced outperformed any other phrase mining methods we studied: TurboTopics, KERT and ToPMine. In addition, SegPhrase+ does not require any kind of NLP methods, thus providing the possibility of extending RecAuth to other languages in the future. SegPhrase+ follows a four (4) step process when mining quality phrases, which are: 1. Frequent Phrase Detection; 2. Phrasal Quality Estimation based on popularity, concordance, informativeness and completeness; 3. Phrasal Segmentation; 4. Feedback as Segmentation Features. SegPhrase+ also incorporates a very small set of user specified labels to perform training. Similar to TopMine, SegPhrase+ uses rectified frequency to perform popularity counting. A score between 0 and 1 is assigned to each phrase mined. With all the steps mentioned, high quality phrases are mined from a given set of text corpora.

2.1.2 Similarity Measures

Similarity measures are also explored in our research to calculate similarity between input keywords and key phrases that represent papers. In our research, we studied Jaccard Coefficient, Cosine Similarity and PathSim.

Jaccard Coefficient calculates similarity between finite sets of samples, and the main principal is to calculate how much overlap intersection of the input sets over the union of input. The larger this overlap is, the more similar we consider the input key phrases are.

Cosine Similarity measure is inspired by using Euclidean dot product to calculate similarity. In order to convert text documents to a vector form, We can leverage Text Frequency (TF) and Inverse Document Frequency (IDF) to convert text streams to vectors.

PathSim is used to perform similarity measurements in heterogeneous network over different meta paths[10]. The main intuition of PathSim is that it explores different types of paths in the heterogeneous network, and the number of paths that connect the two components. The result PathSim produces favors peers other than a hierarchical relationship.

2.2. Related Work

There are some existing systems that make citation recommendations based on queries. ClusCite is an algorithm that recommends relevant papers to cite given a set of query manuscripts. It greatly leverages the nature of heterogeneous network, and explores different meta paths to perform recommendation. The framework makes the assumption that instead of hard-clustering citations, soft clusters of different interest groups would produce better results. Having obtained the different interest groups, the framework predicts citations by using separate models for each group. More specifically, the framework uses group membership information and infers authority and relevance of each group. The authority scores are propagated via iterations, which is a joint optimization problem[11]. Though the system is effective at recommending papers, it is not capable of recommending authors.

Many other citation recommendation systems also exist, many of which depend on different types of information, such as paper content, known citations, paper venues, and paper authors. A particular effective technique, developed by Yu *et al.*, extracts meta-path based features from heterogeneous bibliographic networks[15]. This technique is capable of capturing text-based similarity, conceptual relevance, and different types of social relatedness. Using this information, accurate citation recommendation can be achieved.

Another system, developed by Bethard *et al.*, utilizes a linearly weighted model that considers both relevance and authority features[2]. This approach addresses the problem that critical information such as the paper importance and quality not being considered.

Authority ranking on graphs is a critical step for such recommendation systems and has been studied extensively. A system, developed by Sun *et al.*, propagates paper authority scores bibliographic networks by considering the paper citation frequency and the prestige of the published venue[12]. If some supervision can be applied to the ranking process, the performance can usually be improved. These methods, however, do not consider the authority bias involved with changes in query topic or interests[1, 6].

A personalized PrankRank algorithm, developed by Haveliwala *et al.*, derives authority scores specifically for each query by considering query topics[7]. This idea allows the representation of different classes using features obtained from object relative authority, which is employed when performing clustering and classification in heterogeneous information networks[8, 14].

3. Framework

The algorithm we developed followed two main intuitions.

1. Use papers to link input keywords and authors

   Since there is no direct correlation between authors and the input set of keywords, we used papers the authors published to link the authors and the input set of keywords. Thus similarity between authors and the input set of keywords can be generated by calculating similarities between papers this specific author published and the set of keywords. Techniques on how to calculate similarities between papers and the set of keywords will be further illustrated in the following sections.

2. Authority score of authors can be inferred by venues this author publish papers at and his co-authors

   The authority score of authors can be divided into two components. The first component is based on the authority score of venues this author publishes papers at. A highly reputable author should publish papers at highly ranked venues, and highly reputable venues attract highly ranked authors. The second component of an authors authority score comes from other authors this particular authors co-authors with. The rank of an author can be enhanced if his co-authors are highly reputable as well. Thus an author's authority score can be calculated by integrating the authority score from venues and authority score from his co-authors.

The algorithm we implemented can be divided into two stages. The first stage is an offline process. The output of this stage is independent of the input keywords or any user specified parameters. The offline stage involves running SegPhrase+ for all the abstract from papers in the DBLP dataset. The second step of the offline stage is used to generate relationship matrices between Author Paper Venue and Author Paper Author. The online process includes calculating similarities between input keywords and key phrases from papers in the dataset. The second step of the online process is to update similarity matrices between Author Paper Venue and Author Paper Author based on the similarity scores calculated.

3.1. Offline Stage

3.1.1 Represent Paper Abstracts Using SegPhrase+

We selected SegPhrase+ among many phrase mining techniques to mine key phrases from the abstract of DBLP papers. SegPhrase+ was selected due to its minimal supervision requirement and its capability to mine high quality phrases. In addition, the scope of SegPhrase+ can be applied to a large text corpora, mining phrases from just the abstract section would result in very short running times. SegPhrase+ does not use any Natural Language Processing (NLP) methods, which allows it to be applied to text written in many languages.

After running the abstract section of each paper through SegPhrase+, we generated a list of key phrases to represent each paper. For example, sthe key phrases generated for the paper Recommending user generated item lists, by Yidan Liu, Min Xie, and Laks V.S. Lakshmanan, are Collaborative Filtering, Model Learning, Collective Matrix Factorization. The phrases generated for the paper Mining high-speed data streams, by Pedro Domingos and Geoff Hulten, are Decision Trees, Incremental Learning, Disk-based algorithms, subsampling. SegPhrase+'s ability to generates very high quality phrases that are representative of the content of papers is highly evident.

3.1.2 Generate APV and APA Relations

APV and APA matrices were generated using papers, authors, and venues in the DBLP dataset. Papers were used to link authors with venues. The intersection between author a and venue v represent all papers a has published at v.

Similarly, an APA matrix is generated from the dataset representing co-author relationships. Each cell within this matrix represents the set of papers two authors have co-authored on. Unlike the APV matrix, this matrix is diagonally symmetric. This is because given authors a and b, a is not related to himself and the a-b relationship is equivalent to the b-a relationship. Sample APV and APA matrices are shown in Figure 1.

3.2. Online Stage

Unlike the offline process, which is directly performed on the DBLP dataset, and is invariant to the input keywords, the online stage will involve calculations of the similarity scores between papers and input keywords, which results in authority scores being updated until convergence is reached.

3.2.1 Generate APV and APA Similarity Matrices

Based on the APV and APA matrices generated and the key phrases extracted from the papers during the

APV	A1	A2	A3	A4	A5	A6	A7	A8
V1	P1 P2 P3	P1 P4	P3	P24	P1	P3	P41	P3
V2	P4 P5	P6	P4 P6	P4 P25	P4 P6	P5 P6	P6	P4 P6
V3	P7	P14 P15	P20	P7	P32	P36	P41 P42	P44
V4	P8	P8	P21 P22	P27 P28	P33 P34	P21 P38	P22	P21
V5	P9 P10	P16 P17	P16	P10	P10	P17	P10	P9
V6	P11	P18	P11	P29	P18	P39	P18	P45
V7	P12 P13	P19	P12 P23	P30 P31	P35	P13 P40	P43	P19

(a) Author - Paper - Venue (APV) relation matrix

APV	A1	A2	A3	A4	A5	A6	A7
A1		P4 P5	P7	P8	P9 P10	P11	P12 P13
A2	P4 P5		P14 P15	P8	P16 P17	P18	P19
A3	P7	P14 P15		P21 P22	P16	P11	P12 P23
A4	P8	P8	P21 P22		P10	P29	P30 P31
A5	P9 P10	P16 P17	P16	P10		P18	P35
A6	P11	P18	P11	P29	P18		P13 P40
A7	P12 P13	P19	P12 P23	P30 P31	P35	P13 P40	

(b) Author - Paper - Author (APA) Relation Matrix

Figure 1: Relation matrices

offline stage, we performed similarity assessment between the input keywords and the phrases. Multiple similarity measures, including Jaccard Coefficient, PathSim and Cosine Similarity, were explored. Cosine Similarity was chosen to be incorporated into RecAuth due to its excellence at representing text similarities and its high efficiency. From the set of key phrases for each paper, we parsed these phrases into unigrams and calculated the similarity score between the input query and the segmented unigrams using Equation 1.

$$\begin{aligned} \text{similarity} &= \cos(\theta) \\ &= \frac{A \cdot B}{\|A\|\|B\|} \\ &= \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \end{aligned} \tag{1}$$

In Equation 1, A_i represents the input keywords and B_i represents the key phrases extracted from each paper.

For each author a and venue v, each cell represents the similarity between the input query and the set of papers a has published at v. The score is simply the sum of similarity scores of all the papers published by a at v with respect to the input keywords. A similarity value of 0 in the matrix either means that a has never published a paper at v or that none of the papers published by a contains any key phrases that are relevant to the input query. A sample APV similarity matrix is shown in Figure 2.

Similar to the APV similarity matrix, the APA similarity matrix is also generated based on the sum of similarity scores between papers co-authored by a and b. Each cell in the matrix represents the similarity

APV	A1	A2	A3	A4	A5	A6	A7	A8
V1	0.0025	0.0000	0.0000	0.2513	0.0351	0.0112	0.0452	0.1367
V2	0.0361	0.0000	0.1326	0.0000	0.2748	0.0000	0.0000	0.2112
V3	0.0000	0.0246	0.0052	0.0000	0.0000	0.0501	0.3198	0.0000
V4	0.3526	0.0000	0.0000	0.0324	0.0221	0.3157	0.0252	0.3170
V5	0.0000	0.2589	0.0913	0.2793	0.0578	0.0000	0.0014	0.7023
V6	0.0124	0.0013	0.0000	0.0000	0.0631	0.0274	0.0251	0.0897
V7	0.3716	0.0892	0.0835	0.0000	0.0000	0.1112	0.1793	0.0000

Figure 2: APV Similarity Matrix

between the input keywords and the papers published collectively by a and b. As shown in Figure 3, all diagonal entries of this matrix are 0 because a has no relation to himself. The other entries with a value of 0 either represents the fact that a and b have never co-authored together or that papers published by a and b do not contain any keywords that are relevant to the words contained in the query.

APA	A1	A2	A3	A4	A5	A6	A7
A1	0.0000	0.2589	0.0000	0.0124	0.0872	0.1127	0.0000
A2	0.2589	0.0000	0.0112	0.0124	0.1217	0.0053	0.0027
A3	0.0000	0.0112	0.0000	0.3068	0.0000	0.0324	0.0215
A4	0.0124	0.0124	0.3068	0.0000	0.0742	0.0423	0.3068
A5	0.0872	0.1217	0.0000	0.0742	0.0000	0.0000	0.0245
A6	0.1127	0.0053	0.0324	0.0423	0.0000	0.0000	0.0000
A7	0.0000	0.0027	0.0215	0.3068	0.0245	0.0000	0.0000

Figure 3: APA Similarity Matrix

These two matrices remain unchanged as we update the authority scores of authors and venues during the iterative process.

3.2.2 Calculate Authority Scores for Both Authors and Venues

Based on the APV and APA matrices generated, we calculate authority scores for both venues and authors. The calculation is based the following rules:

1. Highly ranked authors publish papers at highly ranked venues

   Authority scores of authors can be inferred by authority scores of venues these authors publish at. This rule can be represented with Equation 2.

   $$r_Y(j) = \sum_{i=1}^{m} W_{YX}(j,i) r_x(i) \tag{2}$$

   In Equation 2, Y represents authors and $r_Y(j)$ represents the authority score of a particular author j.

2. Highly ranked venues attract highly ranked authors

Authority score of venues can be inferred by authority scores of authors. This rule can be represented with Equation 3.

$$r_X(i) = \sum_{j=1}^{n} W_{XY}(i,j) r_y(j) \tag{3}$$

In Equation 2, X represent venues and $r_X(i)$ represent the authority score of a particular venue i.

3. Rank of an author is enhanced if it co-authors with many highly ranked authors

   Leveraging the APA matrix, if an author co-authors with authors that are highly reputable, his authority score could be enhanced. This rule can be represented by Equation 4.

$$\begin{aligned} r_Y(j) =& \alpha \sum_{i=1}^{m} W_{YX}(j,i) r_x(i) + \\ & (1-\alpha) \sum_{j=1}^{n} W_{YY}(i,j) r_y(j) \end{aligned} \tag{4}$$

   The first part of Equation 4 is the identical to Equation 2. The second part of Equation 4 is generated from the APA Similarity matrix.

The equations that correspond to the three specified rules were used to calculate the authority scores of papers and venues. Authority scores for venues are calculated directly from the APV Similarity matrix shown in Figure 2 by simply adding all the similarity scores in each row. In order to calculate the authority scores of authors, we need to leverage both the APV Similarity matrix shown in Figure 2 and the APA Similarity Matrix shown in Figure 3. This is a sum of authority score of all the venues an author a has published papers at and all the authors a has co-authored with on any paper.

3.2.3 Update Ranking Until Convergence

Based on the methodology introduced in the previous section to calculate authority scores of authors and venues, we iteratively update authors' authority scores and venues' authority scores. We terminate our iterations when one of the two conditions specified below is met:

1. Maximum number of iterations τ reached

   The user is allowed to specify a cap for the maximum number of iterations to be performed. Increasing the value of τ improves the quality of recommendation at the cost of increasing running time.

2. Rankings of the top k authors remain unchanged over σ iterations

   Since we are only concerned about the top ranked authors, users are given the option to specify the number of recommended authors, k. The authority scores of authors and venues will change during every iteration. However, the algorithm can be terminated if the e rankings of the top k authors have remained consistent for τ iterations.

Algorithm 1 outlines the entire algorithm.

Algorithm 1: RecAuth algorithm

Data: abstracts, venues, and authors of DBLP papers
Input: query keywords, τ, k, σ
Output: top k ranked authors

```
perform key phrase extraction from abstracts;
construct heterogeneous bibliographic network;
construct APV, APA matrices ;
construct APV, APA similarity matrices based on cosine similarity between input keywords and key phrases from abstracts;
calculate authority scores for authors and venues ;
initialize rankings;
repeat
    calculate authority scores for authors and venues;
    rank authors based on the new authority scores;
until maximum iteration τ is reached OR ranking for top k authors remain unchanged for the last σ iterations;
```

4. Experiments

The performance of RecAuth is evaluated. The algorithm was tested on a real-world dataset and the quality of the author recommendation was assessed.

4.1. Data Preprocessing

The experiment performed involved the DBLP dataset[1]. In addition to containing information such as authors, venues and titles for each paper, much of the data contained in this particular set of data also carries each paper's abstract information. Paper data that do not carry abstracts were filtered out.

[1] http://dblp.uni-trier.de/db/

4.1.1 Key Phrase Extraction

Instead of just using the words contained in the title of each paper, which is not always representative of the content of the paper, we used SegPhrase+ to extract keyphrases from the abstract. Combining the extracted key phrases with the existing key words produced vectors that are a fairly complete representation of each paper.

4.1.2 Heterogeneous Bibliographic Network

Using the constructed keyword and key phrase vector, the title, authors, and venues associated with each paper, a heterogeneous bibliographic network was constructed for the entire dataset. The network facilitates the association of authors with venues and co-authors, as mentioned earlier. The schema of the network is shown in Figure 4.

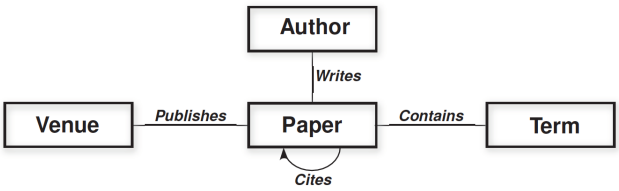


Figure 4: Heterogeneous network constructed from DBLP data

4.2. Experimental Parameters

RecAuth contains many parameters that can be tuned influence the performance. During our experiments, trial and error was performed to determine the values of these parameters that would in optimal recommendation results. We started by setting α value in Equation 4 to 1, which represents that the authority score of each author completely rely on the venues they publish papers at, decremented α by 0.1 during each step, and continued this process until α reaches 0. α score equals to 0 represents that the authority score of an author is constructed entirely based on the people he co-authors with. It was found that setting α value in Equation 4 to 0.6 resulted in reasonable authority ranking scores. The ranking produced by α value of 0.6 is also very consistent with the domain knowledge of highly reputable authors in each field. This means that the venues each author publishes paper at hold slightly more weight than each author's coauthors. This is reasonable because certain conferences, such KDD and ACM, are highly prestigious and only accept truly high quality publications.

The number of iterations for propagating ranking between authors and venues play a large role in the quality of the recommended authors. It was also found that setting the maximum number of iterations τ produced a nice balance between the running time and the quality of the output.

The parameter k simply controls the number of authors recommended, which does not actually affect the overall quality of the algorithm. Setting k to too high a value, however, often defeats the purpose of recommending the most reputable authors. The parameter σ influences the running time of the algorithm quite significantly. Yet a high enough value is required to ensure the algorithm does not terminate too early. It was found that setting k and σ to 10 and 20, respectively, resulted excellent recommendations.

A summary of all parameter values used during the experiments is shown in Table 1.

Table 1: Summary of all parameters used during the experiments

Parameter	Value
α	0.6
τ	500
k	10
σ	20

4.3. Framework Performance

Various key phrases were used as input to test the quality of the author recommendation made by the RecAuth framework. The results were compared against the results returned by entering the identical queries into Google Scholar.

Table 2 shows the result of querying the words "data mining." Our framework returned some of the most well-known authors in the data mining community, such as Jiawei Han, Philip S. Yu, and Christos Faloutsos. Compared to the search result returned from Google Scholar, which is shown in Figure 5, RecAuth's results are considered superior. Although Google Scholar returned some papers whose authors are relevant and reputable, the number is far fewer.

The results for additional queries, "database system" and "reinforcement learning", for RecAuth are shown in Table 3. Similar to the previous query, our framework produced many of the most reputable authors in their respective field of research. The results for Google Scholar, shown in Figure 6, only returned papers that are a very small subset of the most well-known authors.

Table 2: Author recommendation made by RecAuth for the query "data mining"

Rank	Author Name	Score
1	Jiawei Han	2.0175E+28
2	Philip S. Yu	1.6769E+28
3	Jian Pei	1.2093E+28
4	Rakesh Agrawal	1.1215E+28
5	Christos Faloutsos	1.0325E+28
6	Wei Wang	8.1145E+27
7	Michael Stonebraker	6.7353E+27
8	Charu C. Aggarwal	6.6373E+27
9	Haixun Wang	6.3833E+27
10	Divesh Srivastava	5.8780E+27

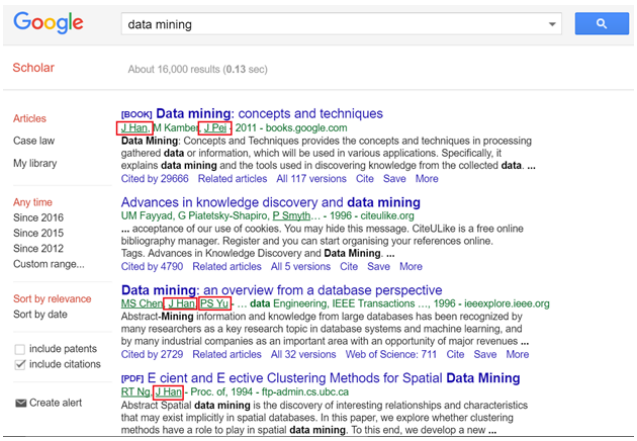


Figure 5: Query result from Google Scholar using the query "data mining'

5. Future Work

Some improvements can be made to further improve the performance of RecAuth. Some possibilities include incorporating Latent Keyphrase Inference (LAKI) to perform paper representation, ranking authors based on more specific disciplines, and allowing user to input paragraphs of text as a query instead of just keywords.

5.1. Incorporating LAKI

LAKI is an framework that can represent documents using a vector of key phrases[9]. Our framework currently makes use of key phrases extracted from abstracts of each papers. There still resides the possibility of certain key phrases being missing from the abstract, which negatively impacts the final result. In addition, phrases generated using SegPhrase+ are treated with equal weights. Leveraging the weighting factor of LAKI could better help us match similar papers to the input keywords. Plus, during the offline stage of LAKI, it leverages the entire text corpora to generate domain key phrase extraction and key phrase silhouetting. It is very likely that incorporating LAKI into RecAuth will dramatically improve the quality of the recommended authors.

Table 3: Additional query results for RecAuth

(a) Query results for the query "database system"

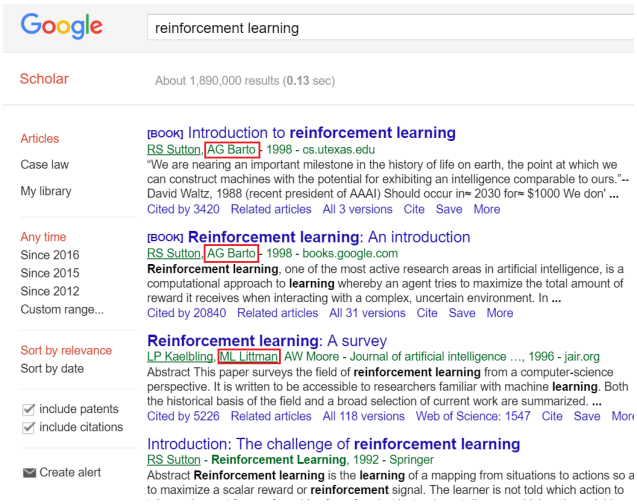
Rank	Author Name	Score
1	Michael Stonebraker	3.5597E+27
2	David J. DeWitt	3.2513E+27
3	Michael J. Carey	2.9365E+27
4	Rakesh Agrawal	2.8426E+27
5	Surajit Chaudhuri	2.8216E+27
6	Divesh Srivastava	2.6091E+27
7	Philip S. Yu	2.4597E+27
8	Michael J. Franklin	2.3284E+27
9	Hector Garcia-Molina	2.2573E+27
10	H. V. Jagadish	2.2515E+27

(b) Query results for the query "reinforcement learning"

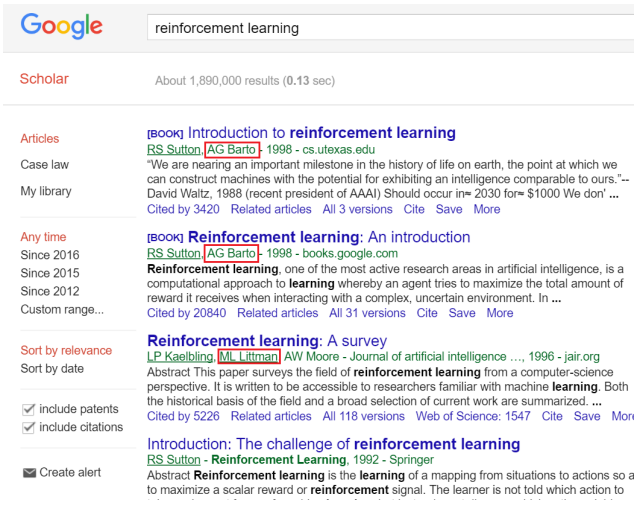
Rank	Author Name	Score
1	Michael L. Littman	5.2443E+14
2	Gerald DeJong	5.0793E+14
3	Andrew G. Barto	4.8060E+14
4	Andrew Y. Ng	4.7524E+14
5	Thomas G. Dietterich	4.5993E+14
6	Shie Mannor	4.2991E+14
7	Prasad Tadepalli	4.2761E+14
8	Sridhar Mahadevan	3.5892E+14
9	Shigenobu Kobayashi	3.4900E+14
10	John Langford	3.0540E+14

5.2. Performing Clustering and Ranking

RankClus is an Expectation Maximization (EM) style algorithm that integrates clustering while performing ranking[13]. In order to better recommend authors specific to a certain discipline, it would be beneficial to subdivide papers into clusters based on each paper's topic. Highly ranked papers from each cluster can then be recommended. RankClus excels in locating authors who are more specific to one cluster. This directly solves the problem of when an author does not



(a) Query results for the query "database system"



(b) Query results for the query "reinforcement learning"

Figure 6: Additional query results for Google Scholar

yet have as many publications, but is still very reputable in his field.

5.3. Allowing Paragraphs as Input Query

RecAuth currently limits the input to a set of keywords that best describes a topic. However, it is possible that a new researcher may not yet have a refined idea of his research. Allowing the possibility for a user to input text paragraphs instead of very concrete keywords could potentially be beneficial. We can leverage SegPhrase+ to mine key phrases from input text paragraphs, which are then used to perform the actual query.

5.4. Automatic Data Update

The dataset used to build RecAuth is static. The actual DBLP data, however, gets updated regularly as new papers get published. An automation technique can be leveraged to perform synchronization periodically. This will ensure up-to-date recommendation results.

6. Conclusion

In this paper, we proposed a method to first find similarity between authors and keywords and then perform ranking on the authors. A framework, called RecAuth, was developed that offers custom author recommendation based on a set of input keywords. By propagating ranking over venues and authors, the algorithm determines the most highly ranked authors among the ones who are similar to the input query. Experimental results show that RecAuth is able to recommend authors that are highly consistent with the results returned by Google Scholar. For certain queries, RecAuth outperforms Google Scholar.

Additional work can be done to further improve the performance of our framework. Terms can be extracted from DBLP papers directly instead of just abstracts so each paper can be more accurately represented by terms. Input queries can be extended to text, such as whole abstracts, instead of just keywords.

References

[1] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 14–23, New York, NY, USA, 2006. ACM.

[2] S. Bethard and D. Jurafsky. Who should i cite: Learning literature search models from citation behavior. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 609–618, New York, NY, USA, 2010. ACM.

[3] D. Blei. Probabilistic topic models. In *Proceedings of the 17th ACM SIGKDD International Conference Tutorials*, KDD '11 Tutorials, pages 5:1–5:1, New York, NY, USA, 2011. ACM.

[4] M. Danilevsky, C. Wang, N. Desai, J. Guo, and J. Han. KERT: automatic extraction and ranking of topical keyphrases from content-representative document titles. *CoRR*, abs/1306.0271, 2013.

[5] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable topical phrase mining from text corpora. *Proc. VLDB Endow.*, 8(3):305–316, Nov. 2014.

[6] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Pro-*

ceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, pages 540–547, New York, NY, USA, 2009. ACM.

[7] 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*, WWW '10, pages 421–430, New York, NY, USA, 2010. ACM.

[8] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1298–1306, New York, NY, USA, 2011. ACM.

[9] J. Liu, X. Ren, J. Shang, T. Cassidy, C. R. Voss, and J. Han. Representing documents via latent keyphrase inference. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 1057–1067, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[10] N. F. Polys, D. A. Bowman, C. North, R. Laubenbacher, and K. Duca. Pathsim visualizer: An information-rich virtual environment framework for systems biology. In *Proceedings of the Ninth International Conference on 3D Web Technology*, Web3D '04, pages 7–14, New York, NY, USA, 2004. ACM.

[11] X. Ren, J. Liu, X. Yu, U. Khandelwal, Q. Gu, L. Wang, and J. Han. Cluscite: Effective citation recommendation by information network-based clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 821–830, New York, NY, USA, 2014. ACM.

[12] Y. Sun and C. L. Giles. Popularity weighted ranking for academic digital libraries. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 605–612, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 565–576, New York, NY, USA, 2009. ACM.

[14] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 797–806, New York, NY, USA, 2009. ACM.

[15] X. Yu, Q. Gu, M. Zhou, and J. Han. *Citation prediction in heterogeneous bibliographic networks*, pages 1119–1130. 2012.