Scholars Recommended System Based On Academic graph knowledge

Project Final Report For EE447: MOBILE NETWORKS, 2017-2018 Spring

Xianze Wu
515030910573

Chengyongxiao Wei
515030910598

Hanyi Sun
515030910569

Supervisor: Prof. Xinbing Wang and Prof. Luoyi Fu
School of Electronic, Information and Electrical Engineering
Shanghai Jiao Tong University

1 Motivation

Most search engines, like Google or Baidu, recommend other items to a user based on the current item the user is searching. Take the following graph as an example, when searching “Kobe” in Baidu, it recommends other basketball players who are related to him (such as entering NBA in the same year or playing in the same team).

Figure 1: recommendations in Baidu

Similar recommendations are also needed in academic search engine. Consider students who have just attended a new school and are looking for mentors, when they search a supervisor on the Internet, other scholars working in similar field and same affiliation may attract them. For researchers who have just entered a specific field, when search a scholar, they might be interested in the associations between this scholar and others, such as his/her co-authors or schoolfellows.

Such recommendations have existed in Google Scholar, it recommends coauthors of the scholars being searched. However, there are much more information in the academic knowledge graphs, such as paper keywords and research fields, conferences and journals where papers are published, etc. It is potential to make use of these information and figure out deeper connections between scholars.

2 Project Description

2.1 Recommender system

A recommender system or a recommendation system (sometimes replacing "system" with a synonym such as platform or engine) is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

Recommender systems have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance, romantic partners (online dating), and Twitter pages.

2.2 Academic Recommendation Based on Acemap and AceKG

Our recommendation system based on an search engine in academic field————Acemap[1]. This search engine provides powerful functions so that we can figure out the relations between authors, papers, affiliations and conferences. Our recommendation system is not just limited to partners or works in the same organization, but also will be dedicated to the research and development of special recommendation system.

Our system is also based on the academic knowledge map – AceKG, a knowledge graph created by the Acemap group from Shanghai Jiao Tong University. (AceKG) describes more than 100 million academic entities and 2 billion and 200 million three tuples information, including about sixty million papers, about fifty million scholars, more than 50000 research fields, and nearly twenty thousand academic research institutions, and the data set is nearly 100G.

Knowledge Graph is a series of different graphics that display the relationship between knowledge development process and structure. It describes knowledge resources and their carriers by visualization technology, mining, analysing, constructing, drawing and displaying knowledge and the interrelation between them. By combining the theories and methods of Applied Mathematics, graphics, information visualization, information science and other disciplines, the methods of citation analysis and co present analysis are combined, and the core structure of the subject, the history of development, the frontier and the overall knowledge structure are displayed with the visual atlas, and the field of knowledge is revealed. The law of dynamic development provides practical and valuable reference for research and decision making.

[1]Acemap main page: http://acemap.sjtu.edu.cn/mainpage

The Knowledge Graph, AceKG, has three advantages compared to other existing academic knowledge map:

- AceKG provides academic heterogeneous atlas, which contains a variety of academic entities and corresponding attributes, and can support a variety of academic data mining topics, such as many topics of the present stage of isomerization of heterogeneous networks.
- AceKG provides an overview of the entire academic circle from a higher point of view, providing a data set of nearly 100G size, including papers, authors, fields, institutions, periodicals, conferences, alliances, and support for authoritative and practical academic research.
- AceKG is given in a structured Turtle file format to reduce the inconvenience of data preprocessing, and is easier to machine and support all Apache Jena API.

As Figure 2 shows, AceKG provides rich attribute information for each entity, and the semantic information is added to the network topology, which can provide comprehensive support for a large number of large academic data mining projects.

Based on the powerful Knowledge Graph, we have done some further work on this. We successfully dug out the latent relationships.

3 Related Work

Recommendation systems typically produce a list of recommendations in one of two ways – through collaborative filtering or through content-based filtering (also known as the personality-based approach)[1]. Collaborative filtering approaches build a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties. These approaches are often combined.

For recommendation system in academic field, a well-known example is Google Scholar, which recommends coauthors as shown in Figure 3. However, a obvious limitation is it only utilizes coauthors information in academic knowledge graph, while much valuable information, research fields, affiliations and conference, etc, stands unused. Another example is a real-time recommendation system for co-author developed by [2]. It applies betweenness centrality of authors in cooperation network, and the overlap of expertise (keywords) between individuals are computed to express their similarity. However, this system based on a small dataset which only comprises less than 10,000 entities, while there are over 100 million entities in the AceKG.

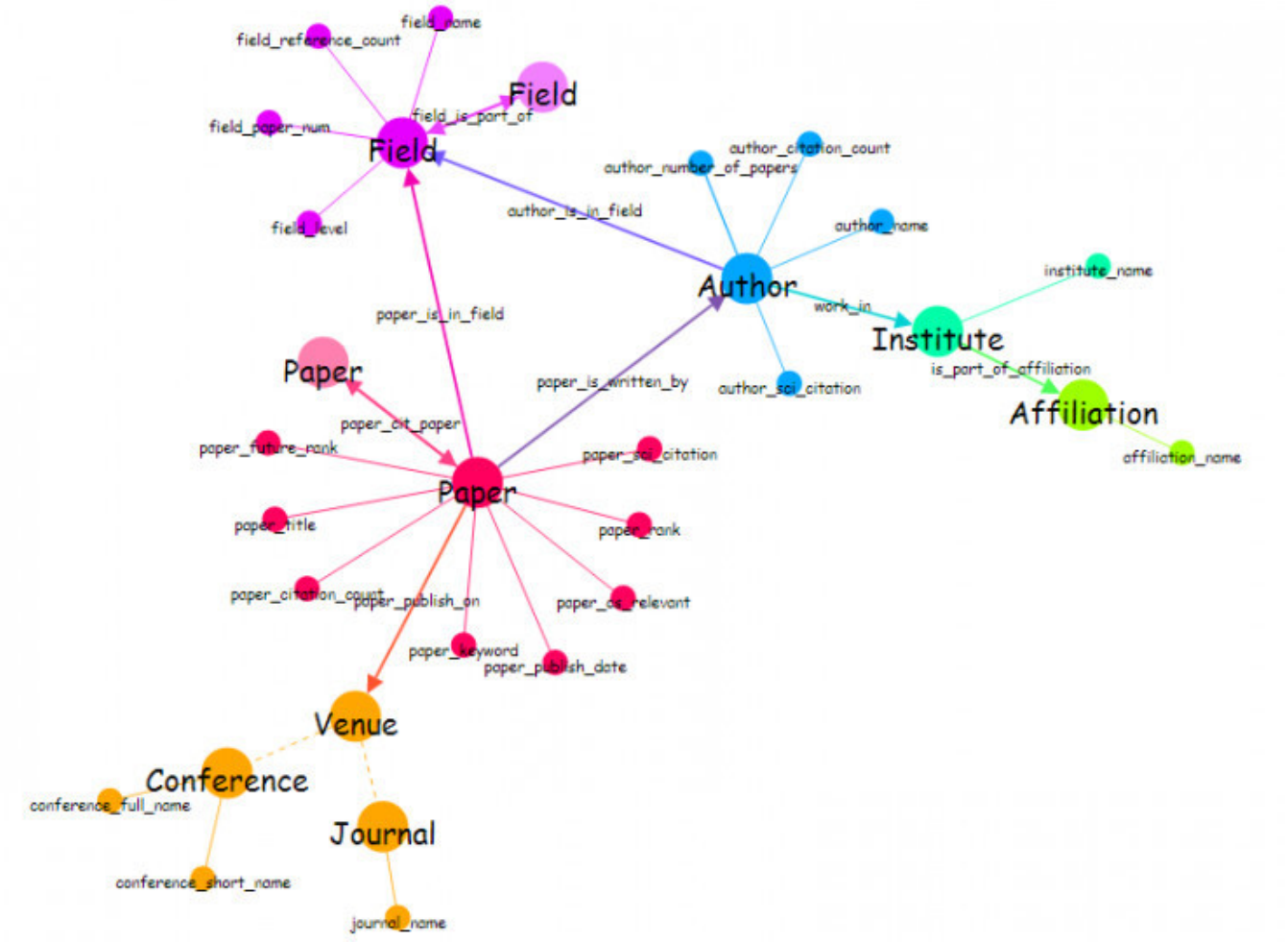

Figure 2: AceKG Structure

What we pursue is a real-time scholars recommendation system which recommends other scholar according to the scholar being searched. In the next section, we will demonstrate how we make use of those significant information.

4 Proposed recommendation algorithm

We consider recommendations based on several dimensions: cooperation between authors, research fields, affiliation and conference. Details of our algorithm are as follows.

4.1 Recommendation based on cooperation network

From our perspective, cooperation network is like social network in academic field. We expect the recommendation show the connections for an author. To dig out latent connection between

Figure 3: Recommendation in Google

two scholars, we consider two kinds of cooperation. One is they are coauthors, which we denote by direct cooperation. The other one is they don't cooperate directly but both of them have collaborated with another scholar, which we denote by cross-author cooperation. This case is quite common in reality. Consider two scholars working in the same lab and have similar research, such as natural language processing. No doubt that they are close related. However, they might don't collaborate frequently as difference between their research focus. However, both of them cooperate frequently with the leader of the lab, so cross-author cooperation can express their close relationship in some extent.

We compute direct cooperation degree DC and cross-author cooperation degree CC between two scholars. For author a and author b, $DC_{a\ to\ b}$ is proportional to cooperation time between a and b after being normalized. When computing their cross-author cooperation degree, we find all agency scholars between a and b satisfying following two conditions:

(1) The scholar has published paper with a, while b isn't an author of the paper.

(2) The scholar has published paper with b, while a isn't an author of the paper.

Take the set of all agency scholars as C. For any scholar $c \in C$, ct_{ca} represents the number of cooperation between a and c satisfying condition 1, ct_{cb} represents the number of cooperation between b and c satisfying condition 2, both being normalized and smoothed. Then, $CC_{a\ to\ b}$ is computed by

$$CC_{a\ to\ b} = sum_C^c ct_{ca} * ct_{cb} \tag{1}$$

Finally, the cooperation degree between a and b is computed considering both $DC_{a\ to\ b}$ and $CC_{a\ to\ b}$. Scholars who have higher cooperation degree with the given author are recommended.

4.2 Recommendation based on research fields

We recommend important scholars in research fields of a given author. From our perspectives, significant scholars often do some pioneer work for this field. Their jobs can help users get familiar with this field quickly. Since the given author usually has several research fields, we focus on scholars who are important in several fields or crucial in a field. We get significant papers in these fields first, then treats their authors as candidates and compute their importance according to their papers.

4.3 Recommendation based on affiliation

For recommendation based on affiliation, we recommend similar scholars in the same affiliation. One promising way to pursue this goal would be expressing the similarity between scholars by their overlap of research fields.

We propose a varient of TF-IDF algorithm. In detail, first, we obtain a list C, which contains scholars in same affiliation and similar research fields with the given author. Then, we fetch top 10 research fields of the given author, where he/she has published the most papers. For author $c_i \in C$, a tf-idf vector $v\_i = v_{i1}, v_{i2}, ..., v_{i10} \in R^10$ is computed to express the research fields of c_i. Let f_j denote the jth field in F, v_i is given by:

$$v_{ij} = \frac{c_i{'s\ number\ of papers\ in\ field\ f_j}}{c_i{'s\ total\ number\ of\ papers}} \tag{2}$$

For each $f\_j \in F$, idf value is given by:

$$D_{f_j} = log(\frac{numbers\ of\ scholars\ in\ C}{number\ of\ scholars\ in\ C\ and\ field\ f_j}) \tag{3}$$

where $D = D_1, D_2, ..., D_{10} \in R^{10}$.

So we can calculate a TF-IDF vector T_i for each $c_i \in C$ by multiplying v_i and IDF. Similar method is applied for calculating the TF-IDF vector of the given author, denoted the vector by T_a. Next, we compute the cosine similarity between T_i and T_a to represent the overlap of research fields between scholar c_i and the given author.

For clarification purposes, more details of the algorithm are as follows

Algorithm 1: recommendation algorithm based on affiliation

Input: the given scholar Au

Output: recommendation list of scholar in same affiliation of the given author based on research fields

$F \leftarrow$ top 10 research fields where Au published the most papers;
$S \leftarrow$ all scholars in same affiliation of Au;
$C \leftarrow$ empty list;
for *scholar* $s \in S$ **do**
    if s *has similar fields with* Au **then**
        push s into C;
    end
end
$m \leftarrow$ len(C);
$n \leftarrow$ len(F);
$tf\_au \leftarrow$ empty list /* tf vector for the given scholar */;
$tf_peers \leftarrow$ empty matrix[m*n] /* tf vector for each scholar $\in C$ */;
$idf \leftarrow$ empty list /* idf for fields in F */;
for $j = 1$ *to* n **do**
    $field \leftarrow$ F[j];
    $tf\_au[j] \leftarrow Au$'s number of papers in $field$ over Au's total number of papers;
end
for $i = 1$ *to* m **do**
    $peer \leftarrow C[i]$ /* ith scholar $\in C$ */;
    $paperTotalCount \leftarrow$ total number of papers of $peer$;
    for $j = 1$ *to* n **do**
        $field \leftarrow F[j]$ /* jth field $\in F$ */;
        if $peer$ *is in* $field$ **then**
            $paperFieldCount \leftarrow$ number of papers of $peer$ in $field$;
            $tf\_peers[i][j] \leftarrow paperFieldCount/paperTotalCount$;
            idf[j] += 1;
        else
            $tf\_peers[i][j] \leftarrow 0$;
        end
    end
end
for $j = 1$ *to* n **do**
    $idf[j] \leftarrow log(m/(idf[j] + 1))$;
end
$score \leftarrow$ empty list; **for** $i = 1$ *to* m **do**
    $score[i] \leftarrow$ cosine similarity between tf_au and $tf\_peers[i]$;
end
return scholars with higher scores;

4.4 Recommendation based on conference

Another recommended method is based on venue. From our perspectives, two scholars have closer contact if they published papers in same venue. We first count at which conferences an scholar published papers in last 10 years. Then we choose the top N venues an author published papers most, and find 20 scholars who published most papers in recent years for each venue. For given $20 * N$ scholars, we calculate scholar R's score:

$$Score_R = \sum_{i=1}^{N} (\frac{s_i}{\sum_{i=1}^{N} s_i} R_i) \tag{4}$$

where s_i means the number of papers a scholar published in i_{th} venue. R_i means the number of papers scholar R published in i_{th} venue. For our project, we set N to 5.

To make our result more valuable, we filter scholars based on citations of their papers. Only choose scholars whose citations more than 1500 and count the number of their papers published in each venue. After that, we got a txt contains 1,990,389 informations for more than 200 thousands scholars. By these information, we obtain a static recommendation for scholars from Shanghai Jiao Tong University.

Unfortunately, there are thousands of scholars published more than ten thousands papers at a normal venue. The time delay to get a list for venue recommendation is too long. It seems not possible for us to make it online, so we just give static recommendation for scholars from Shanghai Jiao Tong University

5 System Developing

After implementing our recommendation algorithm, we add it to the website of Acemap. Basically, we change the author page of Acemap.

Our goal is an on-line recommendation system, so we have to consider several engineering problems. The most difficult problem is executing time. Time-consuming algorithm is unacceptable for users. However, in academic knowledge, an entity may connect to such a huge amount of other entities. For example, a scholar have published hundreds of papers. Thus, as distance between two entities grows, the number of possible connections between them grows exponentially, and computation cost increase in similar speed. To speed up our algorithm, we prune search results based on features of entities as described in section 3, like setting threshold for an author's number of papers. Although it seems that these methods are quite simple and intuitive, they significantly reduce the running time.

Besides, we must guarantee the robustness of system. Various exceptions may occur when the system runs. It takes us so much time to test our system and detect potential bugs as possible as we can.

6 Result

In other recommendation systems, such as commodity recommendation in Tmall[2], recommendation results are evaluated by users' click rate. Unfortunately, there is no such data in Acemap, so we have to find alternative methods. From our perspective, opinion about recommendation results of the scholar being searched are results is valuable for evaluating and improving our results, because results are obtained according to his/her information in academic knowledge graph. Inspired by the opinion, we showed our result to some professors such as Prof. Xinbing Wang and Prof. Weinan Zhang, and improving our algorithm by their comments. Besides, effectiveness of our cross-author cooperation idea is evaluated by a case study. Demo page is also provided.

6.1 Case study for cross-author cooperation

Figure 4 is the recommendation result Prof. Xinbing Wang as a case study for "cross-author" cooperation.

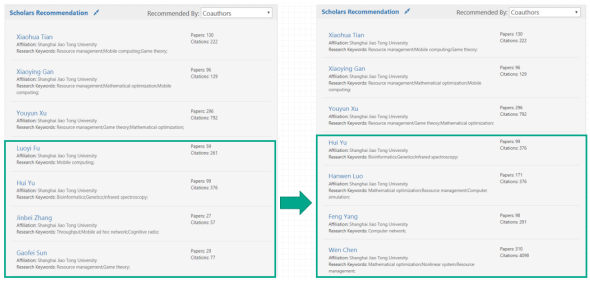

Figure 4: results without or with cross-author cooperation

The left part is result without cross-author cooperation, only according to co-author times. The right part is result with co-author times and cross-author cooperation. As shown in the graph, the system recommends some news scholars after considering cross-author cooperation, like prof 罗汉文 and 杨峰, both of them are members of Prof. Wang's lab(IIoT). They don't cooperate quite frequently with Prof. Wang but do closely relate to him.

[2] https://www.tmall.com/

6.2 Demo page

Final user interfaces and recommendation results are as following graphs. Besides, demo page for Prof. Xinbing Wang and Prof. Luoyi Fu are available on:

Demo for Prof. Xinbing wang

Demo for Prof. Luoyi Fu

In common cases, recommendation will be shown in several seconds. However, sometimes the server and database which recommendation system based on are unstable. So if you haven't see the result after a period of time(like 10 seconds), refreshing the page may help. We feel sorry about that but it's something beyond our control.

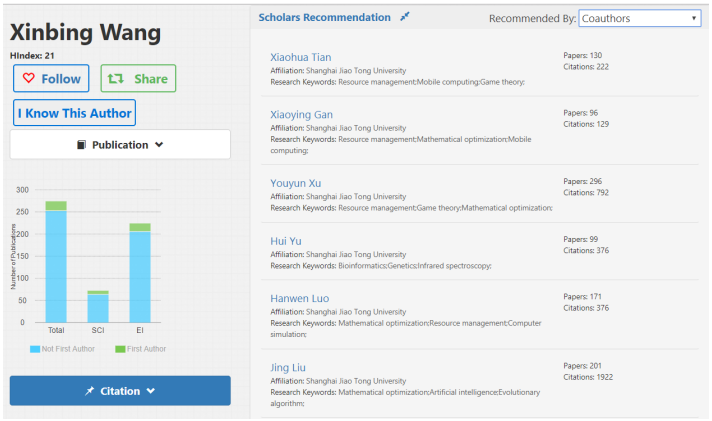

Figure 5: recommendations by coauthors

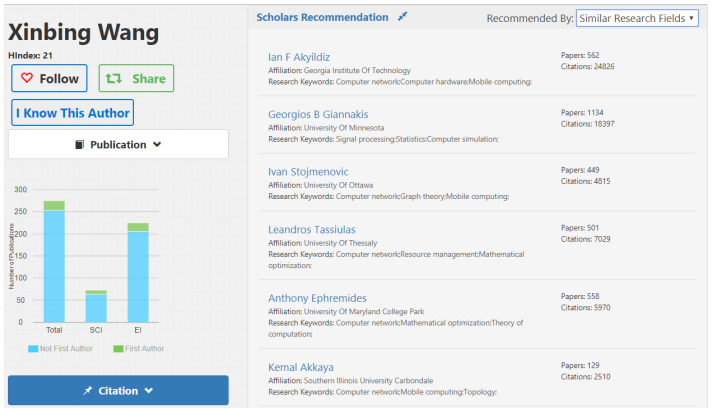

Figure 6: recommendations by similar research fields

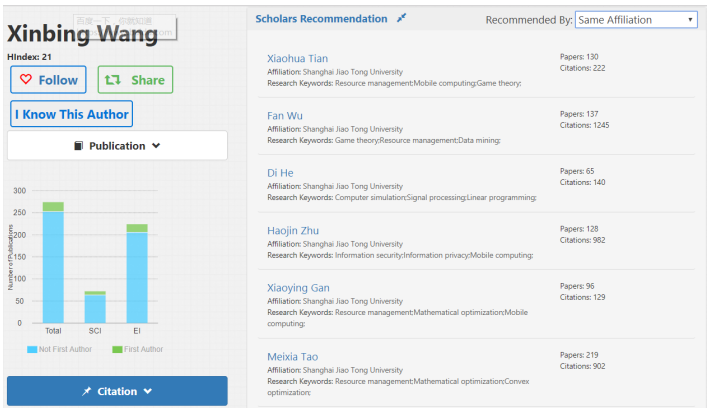

Figure 7: recommendations by same affiliation

7 Conclusion

In our project, we designed a real-time recommendation system based on academic knowledge graph, which recommend scholars in three dimensions: cooperation network, research fields and affiliations. Besides, we finished offline recommendation based on conference.

In the future, we intend to optimize our recommendation algorithm to obtain better results which benefit users much more. In addition, it's such a pity that there is not much user data in Acemap now. If we get more user data, we will improve our algorithm for not only recommendations when searching a scholar but also personalized recommendations for specific users.

8 Task division

Xianze Wu: recommendation algorithm based on research fields and affiliation, apply algorithms to Acemap, UI design

Chengxiaoyong Wei: recommendation algorithm based on cooperation network, apply algorithms to Acemap

Hanyi Sun: recommendation algorithm based on conference, UI design.

References

[1] F. O. Isinkaye, Y. O. Folajimi, and B.A. Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273, 2015.

[2] Rory L. L. Sie, Hendrik Drachsler, Marlies Bitter-Rijpkema, and Peter Sloep. To whom and why should i connect? co-author recommendation based on powerful and similar peers. *International Journal of Technology Enhanced Learning*, 4(1/2):121–137, 2012.