A Method of same name disambiguation towards literature search

Pengfei Lei, Yukun Li*, Pengcheng Tian, Li Liu, Dexin Zhao
Tianjin University of Technology, 300384
Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin, China
Key Laboratory of Computer Vision and System, Ministry of Education Tianjin, China
superlpff@163.com, liyukun@tjut.edu.cn, pengchengtian@126.com, superliuli@163.com, zhaodexin@tjut.edu.cn

Abstract—The issue that many authors share the same name results in many problems for paper search and the study of the Scientists Collaboration Network. In this paper, we study the problem of name ambiguities towards author-based search, propose a framework on same name disambiguation towards literatures searching, and propose a method to classify author-based searching results based on researchers of the real world. The experimental results show the effectiveness of the methods we proposed.

Keywords—Same Name Disambiguation; author search; Classification;

I. INTRODUCTION

With the development of internet, many scholarly digital libraries (DLs) appear, such as DBLP, CiteSeer, PubMed, ACM DL, IEEE DL, and so forth, they facilitate literature search and provide researchers good datasets to study the scientists collaboration network. But how to utilize the collections of literatures for research is still a challenging problem. According to Lee et al. [1], the challenges to have high quality content comes from data-entry errors, disparate citation formats, lack of (enforcement of) standards, ambiguous author names, and abbreviations of publication venue titles, and etc. Among these challenges, author name ambiguity has drawn a lot of attentions from the DL research community due to its inherent difficulties.

In the real-world, there is a motivation scenario. In DBLP, when people want to find some papers by inputting a researcher's name, she/he often get a large number of similar author names, which often makes the user be in a dilemma. Figure 1 shows the results returned by DBLP when inputting author name "Dong Xin". Figure 1 only lists a part of author names. The total number of the returned author names is 43. Facing the author names, the user will face two problems: (1) In fact, according to our knowledge, "Xin Dong" and "Xin Luna Dong" are the same person of the real world, although they are listed as two items; (2) when people select " Dong Xin ", a great number of papers authored by " Dong Xin " will be returned, although the people possibly point to different person of the real world. From the example we can see, the problem of same name disambiguation is still a problem which is not solved well.

A. Related Work

In fact, the problem of name ambiguities has been noticed for a long time. Researchers have made great efforts to attack the problem by trying a lot of methods. Lizhu Zhou et al.[2] proposed effective framework named GHOST (abbreviation for GrapHical framework for name disambiguation). Tang Jie et al.[3] formalized the problem in a unified probabilistic framework, which incorporates both attributes and relationships. Wu Jiang et al.[4] proposed a recursive reinforced name disambiguation method that integrates both co-authorship and affiliation information, especially in cases of scientific collaboration and mobility. Yang Xia et al.[5] presented an approach to the Chinese Personal Name Disambiguation. Stasa Milojevic[6] derived realistic estimates for the precision of simple-initials-based methods using simulated bibliographic datasets in which the true identities of authors are known. Ferreira et al.[7] proposed a novel two-step disambiguation method, SAND (Self-training Associative Name Disambiguation). Yoshida[8] proposed to use a two-stage clustering algorithm by bootstrapping to improve the low recall values, and implement a system that disambiguates person names in Web search results. Byung-Won On et al.[9] proposed two scalable graph partitioning algorithms known as multi-level graph partitioning and multi-level graph partitioning and merging to solve the large-scale name disambiguation problem. Pei Li et al.[10] studied linking temporal records, and proposed clustering methods that considering the time order of the records and making global decisions.

Figure 1 The authors returned by DBLP when searching "Dong Xin"
Although there are some works about author name disambiguation. Because of the inherent difficulty of this problem, the problem is still not solved well.

In this paper, we focus on proposing a lightweight online method to the problem of same name disambiguation in the case of literature searching.

B. Contribution Summary

Our main contributions can be summarized as below.

• Propose a framework on same name disambiguation towards literatures searching. It considers not only the problem of "many person share one name", but also the problem of "many names match one person".

• Propose the three-stage method about disambiguation of same name. The first is to generate a possible name set based on the input author name; the second is to generate searching results of literatures based on the author name set. The third is classification of searching results.

The rest of the paper is organized as follows: In Section 2, we define the problems. In Section 3, we propose classification method based on multi-attribute. Section 4 is about the experiments. In Section 5, we conclude this paper.

II. PROBLEM DEFINITION

In this section, we will introduce the reasons of author identification, the definition of the problem, and the solution framework.

Author identification includes two meanings: multiple people share one name, and many name point to one person.

The problem focused on by this paper is that: For a data set of papers, when people input a researcher name, all the papers whose author names include the given name will be returned, and the results will be classified based on the researchers sharing the input name.

It can be formulized as below: Let N be the author name input by people, S be a set of papers whose authors' name include the author name N, and S is denoted as a set \{Pi|1<i<M\}, where Pi is a paper, and M is the number of the papers of S. A paper is represented as a 4-tuples (Ti,Ab,Jc,Au), where Ti means title of a paper, Ab is the abstract, Jc is where the paper is published (the name of journal or conference). Because one paper is often authored by more than one people, we describe the element Au as a set of 2-tuple (Na, Af), where Na means the corresponding name and Af means the affiliation of the author.

Another problem focused on by this paper is to classify the papers pointed to the same author into some subsets. To each subset, the same-name authors will be the same people of the real world. Besides, some different author names may correspond to the same person of the real world, so their papers should be put into one class.

For example, let \{p1, p2, p3\} be three papers returned by searching “Xin Dong” and \{p4\} be the paper returned by searching “Xin Luna Dong”. In fact, p1, p2 and p4 are authored by the same person and p3 is authored by another person. The aim of this paper is to find a method to divide the searched papers into several classes, and each class point to one person of the world.

III. CLASSIFICATION METHODS

To solve the problems mentioned in section II, in this section, we propose an effective solution which includes three steps: Generate a set of initial paper set, get subclasses based on authors' affiliations, cluster based on co-author and papers publication information.

A. Generate a Set of Initial Paper Set

To make the searching results more complete, we firstly need to have all possible names who may map the input name. Our approach of obtaining an initial set of same-name authors is that: when given a name, we firstly get a candidate name set where each name is a possible form of the given name. The main problem is that authors have middle name in some papers, but there isn’t in other papers.

We propose the following method to generate possible author sets based on user input: Let “X (Y) Z” be the author name input by user, where Y is omissible, (1) if Y is omitted, and the possible author set is (“X Z”, “Z X”, “Z * X”, “X * Z”), where “*” means the middle name can be any string;(2) if Y is not omitted, the possible set is (“X Y Z”, “X Z”, “Z X”). For example, if the given name is “Xin Luna Dong”, we will get a name set as below (“Xin Luna Dong”, “Xin Dong”, “Dong Xin”). If the given name is “Xin Dong”, the possible name set is (“Xin Dong”, “Dong Xin”, “Xin * Dong”)

Based on the possible author set, we can get the initial paper set. The approach is that to any paper Pi, if the authors of Pi include any one of the possible names, Pi will be taken as one of the initial results. Now, based on the initial set, we propose an approach which includes two steps: (1) Classify the papers into some subsets by strict rules to make sure each subset's same-name author point to the same person; (2) Combine the subsets based on some rules, and get the final classification results.

B. Get Subclasses Based On Authors’ Affiliations

The aim of the approach is to divides the initial paper set into some subsets \{Si\}, where each Si is a set of papers, where the same-name authors are the same person at high possibility. The core question is how to get the subsets. According to experience of people, the people of the same affiliation share same name at low possibility. Then we generate the subsets of paper of same-name author based on affiliation.

We propose a method to find and extract affiliations of authors of DBLP papers, and implement a system named iSearchPapers. For how to get affiliation is not the focus of
this paper, we don't introduce the method in detail. Based on the dataset we collected, we have a set of affiliations referred to by authors of DBLP papers. Therefore, let P1 and P2 be the two papers and N be the sharing author name of them, if the affiliations of name N in P1 and P2 are the same, we regard the two author name N as the same person.

By analyzing, we find the following problems: Affiliations may have different format. One affiliation may have some different representations, which includes abbreviations, and other omitted cases. For example, "University of Washington" can be represented as "Univ. Washington" or "WU", Google can be written as “Google Inc.”. Because of the situations, in the experiments of this paper, we take manual methods to unify the name of affiliations. For example, we take "Google" to represent “Google Inc.”. After that, we can classify papers into subclasses based on the affiliation of the author.

C. Cluster Based on Co-authors and Publication

The subclasses have the following characters: (1) The same-authors in the same subclass should be one person at high possibility; (2) The same-authors in the different subclass are also possibly the same person, because people may change their affiliations. Therefore we propose a method to combine the subclasses which referring to the same person.

Our approach is based on the observations introduced in section II. We consider two attributes in the clustering method: co-authors and publications. We consider the journal name and conference name to represent the research area approximately. Algorithm 1 shows the details of process. The input is a set of subclasses and the output is a set of classes. In this function we take a greedy method to combine the subclasses who possibly point to the same person.

Algorithm 1 Clustering method

Input: A list of subclasses of paper S, M is the number of subclasses

Output: A final classes of paper R

1. procedure clustering process
2.    N = 1
3.    for every subclass s’ of S
4.       for every class r’ of R
5.         if r’ and s’ has overlaps of co-authors or publications
6.             Insert elements of s’ into r’
7.         else
8.             Insert s’ into R as a class
9.         end if
10.    end for
11. end for

IV. EXPERIMENTS

In this section we show the experiments and do some evaluations.

A. Experimental Set

We take parts of well-known literature set DBLP as experimental data set and take some authors as samples for experiments. Table I shows the authors we selected. Table I includes author name, the amount of papers, the time period we set, and the data sources. DBLP.ACM means the papers in DBLP and published by ACM. When evaluating the results of classification, we considered the file names that correspond to the largest number of persons as sample.

We create the baseline with manual methods. The process is as below: To each author name of table I, we selected all papers by inputting the author name. To classify the papers, we try our best to make a just decision by manually checking the authors’ homepages, multiple literature sets like Google scholar, ACM digital library, IEEE digital library, and etc.

<table>
<thead>
<tr>
<th>Author name (Alternate name)</th>
<th>amount</th>
<th>Years</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xin Dong (Xin Luna Dong)</td>
<td>13</td>
<td>1991-2011</td>
<td>DBLP.ACM</td>
</tr>
<tr>
<td>Qing Li</td>
<td>47</td>
<td>1995-2011</td>
<td>DBLP.ACM</td>
</tr>
<tr>
<td>Wei Lu</td>
<td>10</td>
<td>1995-2011</td>
<td>DBLP.ACM</td>
</tr>
<tr>
<td>Wei Wang</td>
<td>107</td>
<td>1996-2012</td>
<td>DBLP.ACM</td>
</tr>
<tr>
<td>Alon Halevy (Alon Y. Halevy)</td>
<td>44</td>
<td>2000-2012</td>
<td>DBLP.ACM</td>
</tr>
</tbody>
</table>

Because the number of the classes generated by our method and the number of the classes of the baseline are not certainly same, therefore we take the following method to create the corresponding relations. Let $C=\{C_1, C_2, ..., C_n\}$ be the classes of papers authored by name N in the baseline we created and $C'=\{C'_1, C'_2, ..., C'_m\}$ be the classes generated by our methods. Then we can get a $m \times n$ matrix $(a_{ij})$, where $a_{ij}$ is the number of the overlapped papers of class $C_i$ and $C'_j$. We take the following steps to decide the corresponding relations.

1. Let $a_{y}$ be the max value of the matrix, we get the following corresponding relation $(C_i, C'_j)$;
2. Cancel the i-th row and the j-th column of the matrix, and from the rest $(m-1) \times (n-1)$ matrix selected the max value and get one corresponding relation;
3. Continue the step (1), (2), until the number of the corresponding relations are equal to min(n,m).

After creating the corresponding relationships, we can evaluate the effectiveness of our method. We take the
popularly-used measures Recall, Precision and F-measure for evaluation. They are computed as blow.

\[
\text{Precision} = \frac{TP}{TP+FP} \quad (1)
\]

\[
\text{Recall} = \frac{TP}{TP+FN} \quad (2)
\]

\[
\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (3)
\]

In the formulas above, TP means the number of the papers classified correctly. FP means the number of False Positive papers. FN means the number of the False Negative papers.

B. The Result of the Experiment

Firstly, we calculate the recall and the precision of each class corresponding to the same name in the classification results. The recall and precision of the same name were the average value of all the corresponding classes. The final recall and precision were determined by the average value of the recall and the precision of all the names in the samples. Table II shows the experimental results in detail.

**Table II Experimental Results of the Persons**

<table>
<thead>
<tr>
<th>Author Name</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xin Dong</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(Xin Luna Dong)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qing Li</td>
<td>0.99418603</td>
<td>0.80232555</td>
<td>0.88804396</td>
</tr>
<tr>
<td>Wei Lu</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Wei Wang</td>
<td>0.91991786</td>
<td>0.85964912</td>
<td>0.8876293</td>
</tr>
<tr>
<td>Alon Halevy(Alon Y.</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Halevy)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From Table II, we can see our method has a good recall, precision and F-Measure totally.

C. Analysis of the Experimental Result

By analyzing of papers which are not correctly classified, we find there are two types of errors. (1) An article was wrongly assigned to a class. (2) A class of articles did not complete.

As to the first problem, after analyzing, we found that the reason was in the step of combining subclasses. According to our method, if two classes of papers have the same journal attribute, they should be merged, so when two same-name people publish papers at the same journal, they would be put into the same classes, although they are not the same person.

The second problem is that the papers of one person are divided into many classes. The papers will have no overlap with the older one if they have changed their affiliations, their papers' co-authors and journals (or conferences).

Besides, after comparing the artificial classification results and the classification results based on the algorithm, we found that the phenomenon that the two results based on different classifications are different existed in some sample name sets. The truth also reflected the limitation of the classification algorithm. In addition, due to the standard of same name disambiguation is that the same-name authors have the same co-authors. However we can’t exclude the situation that the co-authors have the same name but are not the same person.

Therefore, the methods still have some limitations, which will be solved in the future.

V. CONCLUSIONS

This paper proposes a framework on same name disambiguation towards literatures searching, and proposes the method about the disambiguation of same name. The first is to generate possible name sets based on the input author name; the second is to generate searching results of literatures based on the author name sets. The third is classification of searching results. This is just a preliminary work. In the future, we will try to improve the effectiveness of the methods.

ACKNOWLEDGMENT

This research was supported by the Natural Science Foundation of China under grant number 61700227, 61202169, and 61170174; Natural Science Foundation of Tianjin under grant number 11JCYBJC26700.

REFERENCES