Incorporating User Feedback into Name Disambiguation of Scientific Cooperation Network

Yuhua Li*, Aiming Wen, Quan Lin, Ruixuan Li, and Zhengding Lu

Intelligent and Distributed Computing Lab,
School of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, P.R. China
wenaiming@smail.hust.edu.cn, linquan.hust@gmail.com,
{idcliyuhua,rxli,zdlu}@hust.edu.cn

Abstract. In scientific cooperation network, ambiguous author names may occur due to the existence of multiple authors with the same name. Users of these networks usually want to know the exact author of a paper, whereas we do not have any unique identifier to distinguish them. In this paper, we focus ourselves on such problem, we propose a new method that incorporates user feedback into the model for name disambiguation of scientific cooperation network. Perceptron is used as the classifier. Two features and a constraint drawn from user feedback are incorporated into the perceptron to enhance the performance of name disambiguation. Specifically, we construct user feedback as a training stream, and refine the perceptron continuously. Experimental results show that the proposed algorithm can learn continuously and significantly outperforms the previous methods without introducing user interactions.

Keywords: Name Disambiguation, User Feedback, Scientific Cooperation Network, Perceptron, Constraint.

1 Introduction

Name ambiguity is widely existing in academic digital libraries, such as Springer, ACM, DBLP and CiteSeer. For different authors may be thought of as the same author, name ambiguity makes data robust even dirty and lowers the precision of researches based on it.

Name disambiguation is a very critical problem in multiple applications, and it is often desirable to be able to disambiguate author names for many reasons. First, while browsing or searching academic papers, users would be interested to find papers written by a particular author. Second, the resulting disambiguated names can be used to improve other applications such as homepage search and calculating the academic ability of scientists.

* Correspondence author.

H. Wang et al. (Eds.): WAIM 2011, LNCS 6897, pp. 454–466, 2011.

Though lots of work has been involved in name disambiguation, the problem has still not been well settled. Furthermore, User feedback which has been widely studied in many other applications and has good performance, is largely ignored in most researches about name disambiguation. To fill this gap, we leverage user feedback to improve the performance of name disambiguation.

This paper focuses on the problem of assigning papers to the right author with the same name. We investigate into name disambiguation of the scientific cooperation network. The problem is formalized as follows: given a list of papers with authors sharing one name but might actually referring to different people, the task then is to assign the papers to different classifications, each of which contains only papers written by the same person. This paper has three main contributions:

- A approach that incorporates user feedback into the perceptron is proposed to handle the name disambiguation of scientific cooperation network. Two new features and a constraint are drawn from user feedback to help achieve a better disambiguation result.
- User feedback is constructed as training stream to train the perceptron, therefore, perceptron can be refined continuously.
- We conducted the experiment using our approach in a real-world dataset, and the experimental results have proved that our proposal is an effective approach to solve the problem of name ambiguity.

The rest of the paper is organized as follows. In Section 2, we review related works. Section 3 gives name disambiguation a formal definition. In Section 4, we introduce the main idea of name disambiguation using constraint-based perceptron and explain how to incorporate user feedback into the model. Section 5 discusses experimental results, while Section 6 gives the conclusion and future work of this paper.

2 Related Works

In this section, we briefly introduce previous works, which fall into two major aspects: name disambiguation and machine learning methods joining user feedback.

Name Disambiguation: A great deal of research has focused on the name disambiguation in different types of data. [1] addressed the problem of named entity disambiguation in Wikipedia. A heuristic approach in [2] is proposed to author name disambiguation in bibliometrics.

[3] proposed a hybrid disambiguation method, which exploits the strength of both unsupervised and supervised methods. [4] described an algorithm for pairwise disambiguation of author names based on a machine learning classification algorithm, random forests.

Two supervised methods in [5] used supervised learning approaches to disambiguate authors in citations. An unsupervised method in [6] is proposed for name disambiguation in bibliographic citations. Besides, [7] proposed an unsupervised learning approach using the K-way spectral clustering method, they calculate a Gram matrix for each person name and apply K way spectral clustering algorithm to the Gram matrix.

[8] proposed two kinds of correlations between citations, namely, topic correlation and web correlation, to exploit relationships between citations. [9] concentrated on investigating the effect of co-authorship information on the resolution of homonymous author names in bibliographic data. [10] explored the semantic association of name entities and cluster name entities according to their associations, the name entities in the same group are considered as the same entity.

[11] presented a constraint-based probabilistic model for semi-supervised name disambiguation, they formalize name disambiguation in a constraint-based probabilistic framework using Hidden Markov Random Fields. [12] proposed a constraint-based topic model that uses predefined constraints to help find a better topic distribution, and in turn, achieve a better result in the task of name disambiguation.

Machine learning methods joining user feedback: Traditionally, machine learning systems have been designed and implemented off-line by experts. Recently however, it has become feasible to allow the systems to continue to adapt to end users by learning from their feedback.

Clicking data is a kind of invisible user feedback information [13]. The machine learning performance can be improved by employing user's clicking data, and it is widely applied to information retrieval and advertising fields.

[14] proposed a machine learning method to understand the keywords submitted by users. [15] analyzed the user feedback reliability in the Internet. [16] proposed a real-time keywords recommendation algorithm. Matthew Richardson [17] put up a method to predict whether a user will click a ads through analyzing user feedback of clicking ads. [18] proposed a solution for users to directly provide feedback and incorporating the feedback into information extraction.

Most of these algorithms access the user feedback information implicitly. User feedback obtained by this way may have lots of noises and the useful information always hide deeply. With the prevailing of Web2.0 applications, interactive design riches the forms of user feedback. New forms of user feedback contain more abundant information, and also more accurate. To the best of our knowledge, no previous work has directly involved in incorporating user feedback into name disambiguation.

As just mentioned, multiple methods have been raised up to solve name disambiguation, but they all face the problem of low accuracy. Besides, there inevitably exist some mistakes in the result, however, those methods can not correct themselves. In this paper, we employ user feedback to name distinguishing in scientific cooperation network, which can revise the result constantly.

3 Problem Formulation

3.1 Problem Definition

In scientific cooperation network, we here give a formal definition of the name disambiguation problem. As our previous work [19], to formally describe the problem, we define the scientists sharing the same name a as a collection $A = \{a_1, a_2, \cdots, a_i\}$, and we can get a publication set sharing the same author name a and denote it as $P = \{p_1, p_2, \cdots, p_n\}$. Our task is to find the real author of these academic papers and tell them apart, that is, partition the academic paper collection P into small collections $\tilde{P}_1$, $\tilde{P}_2$, ..., $\tilde{P}_i$, make sure each collection $\tilde{P}_i$ only contains papers written by one scientist, as shown in Figure 1.

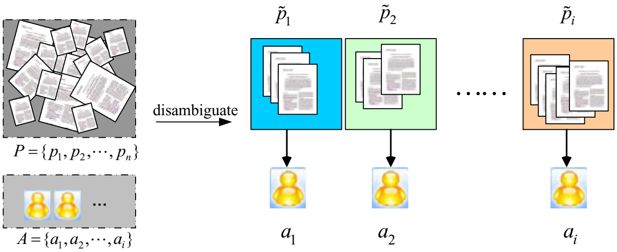

Fig. 1. Name Disambiguation in Scientific Cooperation Network

3.2 User Feedback

The disambiguation result inevitably exists some mistakes which can be easily found out by the users, for example, if one of the target paper's authors sees the disambiguation result about his name, he can easily find out the mistakes.

The original feedback collected from users inevitably contains some mistakes. If the noises of user feedback are not filtered, the performance of name disambiguation will be affected. We divide user feedback into three kinds according to the users providing the feedback.

1. Fully credible user feedback. The feedback is provided by one of the target paper's authors or one of the the target paper's co-authors. As authors or co-authors of the target paper are very familiar with the papers, so their feedback is credible.

2. Credible user feedback. The feedback is provided by the friends of the ambiguous authors. The ambiguous authors' friends can be considered persons who have co-worked with the ambiguous authors. These users are not directly involved in the creation of the target paper, so they are not very sure about who are the authors, there might exist certain misjudgments.

3. Generally credible user feedback. The feedback is provided by users without explicit relationship with the authors. Their feedback is unsure.

3.3 Feedback Training Stream

The biggest challenge of the machine learning based name disambiguation algorithms is the construction of training set. The training sets of the traditional machine learning algorithms are generally static, that is, learning all the training sets at one time. The user feedback arrives continuously, therefore, the training set constructed by user feedback should be provided to the preceptron as a stream for real-time learning.

The stream of training set constructed by user feedback is showed as (1).

$$feedback\_training\_stream = [(p_{i1}, p_{j1}, v_n, c_n), (p_{i2}, p_{j2}, v_n, c_n), \ldots, (p_{in}, p_{j_n}, v_n, c_n)] \tag{1}$$

It is a sequence constructed by user feedback according to the time stamp $\{1 \cdots n\}$. Where the value of v_n is 1 or 0, denotes user feedback considers the two papers are written by one author or different authors respectively, and c_n denotes the reliability of user feedback, if it is fully credible, the value is 1, otherwise if it is credible, the value is 0, and the value is -1 when it is generally credible.

3.4 Feature Definition

A variety of information can be utilized for name disambiguation [19]. First, we define a set of features to be exploited for each paper pair (p_i, p_j) as

$$R = \{r_1, r_2, \cdots, r_k\} \tag{2}$$

where each ri denotes one feature capturing one relationship between papers pi and pj, as shown in Table 1. All the features are defined over the papers sharing the same primary author. For each feature in Table 1, the feature value is binary, that is, if the description is true, then the value is 1; otherwise 0. If the feature value is 1, then the feature indicates that the two papers are probably written by the same author. More detailed descriptions about the features in Table 1 can refer to [19].

Besides the above seven features, two new features are drawn from user feedback. According to the difference of users' credibility, we extract two features from credible user feedback and non-credible user feedback respectively.

Table 1. Feature definition for paper pair(p_i, p_j)

R Feature	Description
r_1Co-Author	exist $u, v > 0, a_i^{(u)} = a_j^{(v)}$
r_2Co-Org	a_i.organization = a_j.organization
r_3Citation	p_i and p_j has citation links
r_4Title-Similarity	a_i.title and a_j.title are similar
r_5Homepage	p_i and p_j appear in someone's homepage
r_6Digital-Lib	p_i and p_j appear in the same Springer/CiteSeer page
r_7PDF File	a_i.organization appears on the PDF format file for p_j and vice-versa

Suppose credible users have returned some user feedback about the paper pair $\{p_i, p_j\}$, of which m users consider the two papers are written by the same target author, while n users consider them belonged to different authors, then feature extracted from such feedback is donated as credible user feedback feature, as shown in the formula (3).

$$r_8 = \begin{cases} \frac{m}{m+n} & \text{if } m+n \neq 0 \\ 0.5 & \text{else} \end{cases} \tag{3}$$

When no one has submitted user feedback, the default value is 0.5.

For the non-credible users, though there contain lots of mistakes, however, when the quantity is larger enough, the correct feedback will dominate, that is, the main opinion of the non-credible users is right. Correspondingly, we can define a non-credible user feedback feature r_9, the formula is the same as (3), but m donates there are m non-credible users considering p_i and p_j written by the same author, and n donates n non-credible users consider the two papers written by different authors. The default value is 0.5 when no one has submitted user feedback.

4 Incorporating User Feedback into Constraint-Based Perceptron

By analyzing the seven features showed in Table 1, we figure out homepage is different from the other features that if its value is 1, we can confirm the two papers are written by the same author, while if the feature value of homepage is -1, we can conclude the two papers are not written by one author. However, we can not have this conclusion with other features. Therefore, we designate homepage as a constraint for the perceptron.

Feedback provided by fully credible users has very high accuracy. Therefore, it can be used as a constraint as the homepage feature, to revise the output of perceptron. The constraint extracted from fully credible user feedback is donated as user feedback constraint, the constraint formula is as (4).

$$Constraint_{feedback}(p_i, p_j) = \begin{cases} 1 & p_i \ and \ p_j \text{ belong to one author} \\ -1 & p_i \ and \ p_j \text{ belong to different authors} \end{cases} \tag{4}$$

The user feedback constraint may be conflicted with the homepage constraint, though the probability is very low. When it happens, we give the priority to the user feedback constraint.

Perceptron with constraint aims to restrict the output. The final output is calculated using the formula (5).

$$output(p_i, p_j) = \begin{cases} 1 & c(y) = 1 \text{ or } c(y) = 0 \text{ and } Sgn(y) = 1 \\ 0 & c(y) = -1 \text{ or } c(y) = 0 \text{ and } Sgn(y) = 0 \end{cases} \quad (5)$$

Where $c(y)$ denotes the constraint, $c(y) = R_0(p_i, p_j)$, $Sgn(y)$ is the output of perceptron and and it is defined as formula (6).

$$Sgn(y) = \begin{cases} 1 & \text{if } \omega * y + b > 0 \\ 0 & \text{else} \end{cases} \quad (6)$$

where y consists of the features showed in Table 1 except homepage and two features drawn from user feedback.

The model of the constraint-based perceptron after importing user feedback is shown in Figure 2. As seen from the graph, two new features r_8 and r_9 are added to the perceptron which are drawn from user feedback. The two features are different from the former several features that their value is not limited to only 1 or 0, it will change according to the user feedback. The appropriate feedback will dominate when there are a great number of user feedback, so the two features actually reflect the major users' view about the name disambiguation result. and the new features will get more accurate as the user feedback multiplied. $Constraint_{feedback}(p_i, p_j)$ drawn from user feedback is utilized to restrict the output of perceptron, so as to make the algorithm more accurate.

The procedures of our proposed method is shown in Figure 3. Firstly, we use the static training set containing only six features to train the perceptron, the output will contain some mistakes, then users find out mistakes and feedback, the feedback will form streams of training set, and also form two new features which will be added to the input of the perceptron. The perceptron revises itself according to feedback training set, updates the weight of each feature and then outputs again, the users will continue to feedback if mistakes are found. This feedback and revise process will continue until no more mistakes are found.

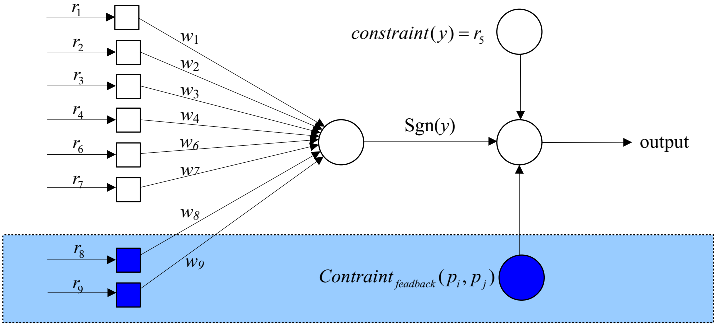

Fig. 2. Incorporating User Feedback into constraint-based Perceptron

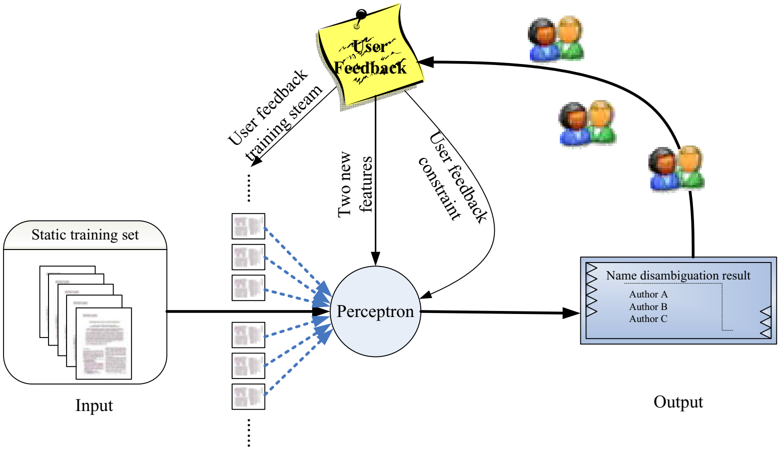

Fig. 3. Procedures of our proposed method

Our algorithm of revising the perceptron with user feedback training stream is as follows.

Algorithm 1. Revising the perceptron with user feedback training stream

```
    Input: User feedback training stream
           $[(p_{i1}, p_{j1}, v_1, c_1), (p_{i2}, p_{j2}, v_2, c_2), ..., (p_{in}, p_{jn}, v_n, c_n)]$
    Output: The final output of the perceptron $y_t$ when no more user
            feedback returned
1   Train the perceptron with static training set, and get the initial weights
    $\omega_t$;
2   repeat
3       Pick a piece of user feedback sample $(p_{in}, p_{jn}, v_n, c_n)$ from the user
        feedback training stream;
4       Calculate the real value of the user feedback sample $\tilde{y}_t = v_n$;
5       Calculate the output of the perceptron $y_t = \omega_t * \mathbf{x}_t$;
6       Update weights $\omega_{t+1} = \omega_t + \alpha_f * (y_t - \tilde{y}_t) * \mathbf{x}_t$;
7   until No more user feedback returned ;
```

In step 5, the two new features drawn from user feedback are added to the $\mathbf{x}_t$. In step 6, the α_f is the learning rate when using user feedback as training set, the value of α_f is related to c_n in user feedback training sample $(p_{in}, p_{jn}, v_n, c_n)$, when the value of c_n is 1, it means that the training sample is fully credible, and in this time, the value of α_f should be greater than the value of α_f sample when the value of c_n in credible sample is 0.

As seen from Figure 3, our approach is differ from previous methods without introducing user interactions that it can correct itself, since it has continuous learning ability.

5 Experiment

In this section, we report our test on the effectiveness of the proposed approach.

5.1 Dataset

To evaluate our algorithm, we create a dataset from four different online digital library data sets: the DBLP, IEEE, ACM, and Springer. This dataset includes 20 real person names with their 1534 papers. For these names, some only have a few persons. For example, ”Juan Carlos Lopez” only represents one person, and ”Koichi Furukawa”, ”David E. Goldberg” and ”Thomas D. Taylor” represent three. However, there are 25 different persons named ”Bing Liu”. To obtain the user feedback and their types, we construct a simple system, one has to register and login to submit his user feedback about the disambiguating results. By this way, we can record his relationship with the author then assign different credibility to the user feedback. Table 2 will give some detail information about the dataset.

Table 2. Number of publications and persons in real name dataset

Name	Pub	Person	Name	Pub	Person
Satoshi Kobayashi	38	6	Bing Liu	215	25
Lei Jin	20	8	R. Ramesh	46	9
David Jensen	53	4	David E. Goldberg	231	3
Thomas Wolf	36	9	Rakesh Kumar	96	12
Koichi Furukawa	77	3	Thomas D. Taylor	4	3
Thomas Tran	16	2	Richard Taylor	35	16
Thomas Hermann	47	9	Jim Gray	200	9
Yun Wang	57	22	Juan Carlos Lopez	36	1
Cheng Chang	27	5	Sanjay Jain	217	5
Gang Luo	47	9	Ajay Gupta	36	9

5.2 Evaluation Measures

We use pairwise measures, namely, Pairwise_Precision, Pairwise_Recall, and F-measure to evaluate the name disambiguation results and for comparison with baseline method. The disambiguation result of paper pairs has two kinds that are written by the same author and by different authors, combined with two kinds of real states. The four states are (1) true positive (tp): paper pairs are written by the same author and the disambiguation result is right; (2) false positive (fp): paper pairs are written by different people while disambiguation think they are written by the same author; (3) true negative (tn): paper pairs are written by different people and disambiguation result also think so; (4) false negative (fn): paper pairs are written by the same author while disambiguation think they are written by different author. The definitions are

$$\text{Pairwise\_Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{7}$$

$$\text{Pairwise\_Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{8}$$

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

Baseline methods. The baselines use the bias classifier learned from each single feature and SA-Cluster which is a graph clustering [20] with the coauthor relationship used as the edge and all the other relationships used as the attribute features, and we use the Pairwise-Classification [19] which combines seven features to the constraint-based perceptron as the no-feedback method for comparison with our approach.

5.3 Experiment Results

The performance of the current algorithm for each author is listed in Table 3 with the performance for baseline method and our algorithm. The results show that our method significantly outperforms the baseline method.

Table 3. Results for 20 real names

	Proposed Method			Baseline Method		
Name	Pre.	Rec.	F	Pre.	Rec.	F
Satoshi Kobayashi	92.05	73.41	81.68	90.12	73.01	80.67
Lei Jin	100	100	100	99.4	99.86	99.63
David Jensen	95.56	87.43	91.31	94.25	87.04	90.5
Thomas Wolf	89.36	33.33	48.55	89.02	31.56	46.6
Koichi Furukawa	96.93	72.72	83.1	96.48	71.59	82.19
Thomas Tran	100	56.04	71.83	99.89	55.87	71.66
Thomas Hermann	100	71.3	83.25	98.89	70.16	82.08
Yun Wang	100	67.65	80.7	99.94	65.1	78.84
Cheng Chang	100	83.95	91.28	98.2	80.96	88.75
Gang Luo	98.41	100	99.2	98.03	97.16	97.59
Bing Liu	88.99	93.88	91.36	88.2	92.46	90.3
R. Ramesh	100	68.12	81.04	99.53	67.46	80.41
David E. Goldberg	99.12	98.26	98.69	99.08	98.08	98.6
Rakesh Kumar	100	97.49	98.73	100	96.43	98.18
Thomas D. Taylor	100	100	100	99.72	100	99.86
Richard Taylor	100	67.82	80.82	99.91	66.76	80.04
Jim Gray	93.76	85.72	89.24	91.5	84.03	87.61
Juan Carlos Lopez	100	89.05	94.21	99.91	87.6	93.35
Sanjay Jain	100	97.74	98.86	99.4	94.16	96.71
Ajay Gupta	100	63.03	77.32	99.75	60.72	75.49

Figure 4 shows the contributions of each feature. For exmaple, Co-Author has very high F-measure because the author names in each paper are complete, compared to the other features. Citation is very useful information because people tend to cite their own papers if they have published related papers. Since the citation information is crawled from the internet, it is not complete, so the recall is low.The precision of homepage is 100%, since the homepage normally contains the owner's papers only. However owners usually only put their best papers on the homepages and not all authors' homepages are available, so the recall is very low. Title Similarity gives very good performance. Because the title information is complete and an author usually publishes a serious of papers in one direction. These authors tend to name their papers in similar ways. More detailed discussions about the contribution of each single feature can refer to our previous work [19].

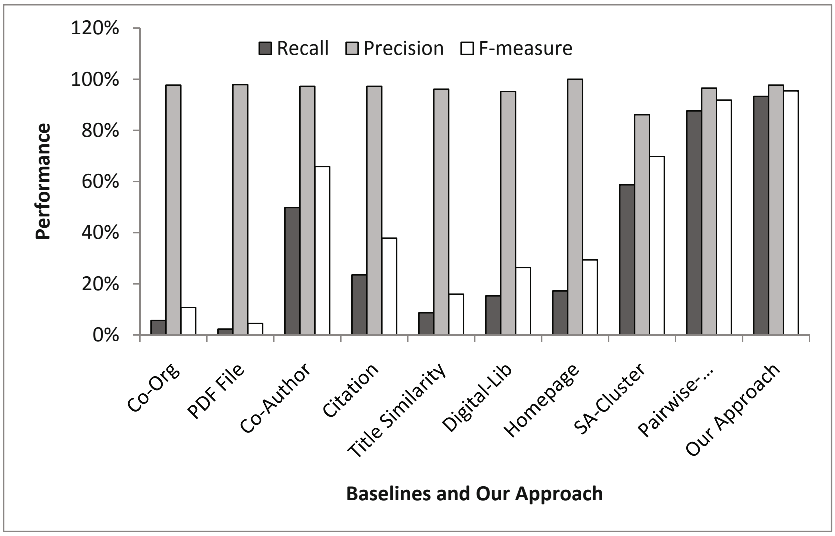

Fig. 4. Comparison of all the baselines and our approach

Specifically, as seen from Figure 4, our proposal significantly outperforms the SA-Cluster method, and despite the inadequate and inaccurate information of user feedback, incorporating user feedback into name disambiguation gives better result in Pairwise_Precision, Pairwise_Recall, and F-measure compared with the no-feedback method Pairwise-Classification.

6 Conclusion

This paper focuses on the problem of name disambiguation in scientific cooperation network. We have proposed a constraint-based perceptron to handle the problem. The method can incorporate features extracted from scientific cooperation network and user feedback to the model. Despite the noises of user feedback, bringing in user feedback gives the best result.

In the future work, we will consider introducing the notion of "credibility" of users, to learn users' credibility dynamically and assign different weights to the feedback according to different credibilities of users, and explore the effects of different kinds of user feedback. Furthermore, we will design more efficient and convenient user feedback forms to attract more users to submit their feedback. And we will consider the case that there is mis-ordering or missing authors in our methods. In addition to the homepage, reading-list will be investigated to further improve the performance of name disambiguation. This model can also be used in other applications for mining the advisor-advisee relationship, searching scientists, etc.

Acknowledgement. This work is supported by National Natural Science Foundation of China under Grant 70771043, 60873225, 60773191.

References

1. Nguyen, H.T., Cao, T.H.: Exploring wikipedia and text features for named entity disambiguation. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS (2). LNCS, vol. 5991, pp. 11–20. Springer, Heidelberg (2010)
2. D'Angelo, C.A., Giuffrida, C., Abramo, G.: A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments. JASIST 62(2), 257–269 (2011)
3. Ferreira, A.A., Veloso, A., Gonçalves, M.A., Laender, A.H.F.: Effective self-training author name disambiguation in scholarly digital libraries. In: JCDL, pp. 39–48 (2010)
4. Treeratpituk, P., Giles, C.L.: Disambiguating authors in academic publications using random forests. In: JCDL, pp. 39–48 (2009)
5. Han, H., Giles, C.L., Zha, H., Li, C., Tsioutsiouliklis, K.: Two supervised learning approaches for name disambiguation in author citations. In: JCDL, pp. 296–305 (2004)
6. Cota, R.G., Ferreira, A.A., Nascimento, C., Gonçalves, M.A., Laender, A.H.F.: An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. JASIST 61(9), 1853–1870 (2010)
7. Han, H., Zha, H., Giles, C.L.: Name disambiguation in author citations using a k-way spectral clustering method. In: JCDL, pp. 334–343 (2005)
8. Yang, K.H., Peng, H.T., Jiang, J.Y., Lee, H.M., Ho, J.M.: Author name disambiguation for citations using topic and web correlation. In: Christensen-Dalsgaard, B., Castelli, D., Ammitzbøll Jurik, B., Lippincott, J. (eds.) ECDL 2008. LNCS, vol. 5173, pp. 185–196. Springer, Heidelberg (2008)
9. Kang, I.S., Na, S.H., Lee, S., Jung, H., Kim, P., Sung, W.K., Lee, J.H.: On coauthorship for author disambiguation. Inf. Process. Manage. 45(1), 84–97 (2009)
10. Jin, H., Huang, L., Yuan, P.: Name disambiguation using semantic association clustering. In: ICEBE, pp. 42–48 (2009)
11. Zhang, D., Tang, J., Li, J.Z., Wang, K.: A constraint-based probabilistic framework for name disambiguation. In: CIKM, pp. 1019–1022 (2007)
12. Wang, F., Tang, J., Li, J., Wang, K.: A constraint-based topic modeling approach for name disambiguation. Frontiers of Computer Science in China 4(1), 100–111 (2010)
13. Wang, C., Han, J., Jia, Y., Tang, J., Zhang, D., Yu, Y., Guo, J.: Mining advisor-advisee relationships from research publication networks. In: KDD, pp. 203–212 (2010)

14. Liu, Y., Zhang, M., Ru, L., Ma, S.: Automatic query type identification based on click through information. In: Ng, H.T., Leong, M.-K., Kan, M.-Y., Ji, D. (eds.) AIRS 2006. LNCS, vol. 4182, pp. 593–600. Springer, Heidelberg (2006)
15. Joachims, T., Granka, L.A., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: SIGIR, pp. 154–161 (2005)
16. Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., Li, H.: Context-aware query suggestion by mining click-through and session data. In: KDD, pp. 875–883 (2008)
17. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: WWW, pp. 521–530 (2007)
18. Chai, X., Vuong, B.Q., Doan, A., Naughton, J.F.: Efficiently incorporating user feedback into information extraction and integration programs. In: SIGMOD Conference, pp. 87–100 (2009)
19. Lin, Q., Wang, B., Du, Y., Wang, X., Li, Y.: Disambiguating authors by pairwise classification. Tsinghua Science and Technology 15(6), 668–677 (2010)
20. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. PVLDB 2(1), 718–729 (2009)