Artif Intell Rev (2014) 41:563–578
DOI 10.1007/s10462-012-9323-5

# Name disambiguation in scientific cooperation network by exploiting user feedback

**Yuhua Li · Aiming Wen · Quan Lin · Ruixuan Li · Zhengding Lu**

Published online: 1 March 2012


**Abstract** Name disambiguation is a very critical problem in scientific cooperation network. Ambiguous author names may occur due to the existence of multiple authors with the same name. Despite much research work has been conducted, the problem is still not resolved and becomes even more serious. In this paper, we focus ourselves on such problem. A method of exploiting user feedback for name disambiguation in scientific cooperation network is proposed, which can make use of user feedback to enhance the performance. Furthermore, to make the user feedback more effective, we divide user feedback into three types and assign different weights to them. To evaluate the effectiveness of our proposed method, experiments are conducted with standard public collections. We compare the performance of our proposal with baseline methods. Results show that the proposed algorithm outperforms the previous methods without introducing user interactions. Besides, we investigate into how different types of user feedback can affect the disambiguation results.

**Keywords** Name disambiguation · User feedback · Scientific cooperation network

Y. Li (✉) · A. Wen · Q. Lin · R. Li · Z. Lu
Intelligent and Distributed Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China
e-mail: idcliyuhua@hust.edu.cn

A. Wen
e-mail: wenaiming@smail.hust.edu.cn

Q. Lin
e-mail: linquan.hust@gmail.com

R. Li
e-mail: rxli@hust.edu.cn

Z. Lu
e-mail: zdlu@hust.edu.cn

## 1 Introduction

Name disambiguation is a very critical problem in bibliographic digital libraries such as DBLP and CiteSeer. These libraries contain a large number of publication metadata records and make these records searchable for academics. However, for there existing multiple individuals sharing the same name, it leads to a tedious task to discriminate among the returned search results between the different people.

Multiple methods have been raised up to solve name disambiguation, but they all face the problem of low accuracy. 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.
- It explores the novel use of user feedback in the problem of name disambiguation. User feedback is constructed as training stream to train the perceptron, therefore, perceptron can be refined continuously.
- We conducted experiments 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. Experiments also demonstrate that different types of user feedback have different effect to the disambiguation results.

The rest of the paper is organized as follows. In Sect. 2, we review related works. Section 3 gives name disambiguation a formal definition. Section 4 describes the form we designed for collecting user feedback in detail. In Sect. 5, we introduce the main idea of name disambiguation using constraint-based perceptron and explain how to incorporate user feedback into the model. Section 6 discusses experimental results, while Sect. 7 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. Nguyen and Cao (2010) addressed the problem of named entity disambiguation in Wikipedia. A heuristic approach in D'Angelo et al. (2011) is proposed to author name disambiguation in bibliometrics. Bekkerman and McCallum (2005) trained a disambiguation support vector machine(SVM) to exploit the high coverage and rich structure of the knowledge encoded in an online encyclopedia. In Minkov et al. (2006), name disambiguation is conducted on email data.

Ferreira et al. (2010) exploited the strength of both unsupervised and supervised methods for name disambiguation. Treeratpituk and Giles (2009) described an algorithm for pair-wise disambiguation of author names based on a machine learning classification algorithm, random forests. Two supervised methods in Han et al. (2004) used supervised learning approaches to disambiguate authors in citations. An method in Cota et al. (2010) is proposed for name disambiguation in bibliographic citations. Han et al. (2005) 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.

Recently, there has been a stream of researches in incorporating the graphical information or context information to enhance name disambiguation. For example, McRae-Spencer and Shadbolt (2006) present a graph-based approach to author disambiguation on large-scale citation networks by using self-citation and coauthor relationships. Tan et al. (2006) present an approach that consider the results of web searches. Chen et al. (2007) proposed an adaptive graphical approach to entity resolution. H Yu et al. (2006) had developed supervised approaches to identify the full forms of ambiguous abbreviations under the condition they appear. Whang et al. (2009) put forward an iterative blocking framework where the resolution results of blocks are reflected to subsequently processed blocks. Wick et al. (2008) proposed a unified approach for schema matching, coreference and canonicalization.

Yang et al. (2008) proposed two kinds of correlations between citations, namely, topic correlation and web correlation, to exploit relationships between citations, in order to identify whether two citations with the same author name refer to the same individual. Kang et al. (2009) concentrated on investigating the effect of co-authorship information on the resolution of homonymous author names in bibliographic data. Jin et al. (2009) 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.

Zhang et al. (2007) 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. Wang et al. (2010b) 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. However, most existing methods ignore user interaction, and the disambiguation precision of the these methods is quite low.

**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 (Wang et al. 2010a). The machine learning performance can be improved by employing user's clicking data, and it is widely applied to information retrieval and advertising fields.

Liu et al. (2006) proposed a machine learning method to understand the keywords submitted by users. Joachims et al. (2005) analyzed the user feedback reliability in the Internet. Cao et al. (2008) proposed a real-time keywords recommendation algorithm. Richardson et al. (2007) put up a method to predict whether a user will click a ads through analyzing user feedback of clicking ads. Chai et al. (2009) 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. For example, plenty of websites for videos require users to add labels to the videos

and rank or score the videos, Google encourages users to re-rank the results returned by its search engine. All these new forms of user feedback contain more abundant information, and also more accurate. To the best of our knowledge, no previous work has directly exploited the user feedback for name disambiguation. Though Davis et al. (2003) has developed an interactive system for users to correct the disambiguation results, however, they do not make use of user feedback to improve the disambiguation results.

As just mentioned, though lots of work has been involved in name disambiguation, the problem has still not been well settled. There inevitably exists some mistakes in the result, as the previous methods do not import user feedback, the mistakes remain uncorrected. In this paper, we employ user feedback to name distinguishing in scientific cooperation network, which can revise the disambiguation results constantly.

## 3 Problem definition

### 3.1 Problem definition

In scientific cooperation network, we here give a formal definition of the name disambiguation problem. As our previous work (Lin et al. 2010), to formally describe the problem, we define the scientists sharing the same name $a$ as a collection $A = \{a_1, a_2, \ldots, a_i\}$, and we can get a publication set sharing the same author name $a$ and denote it as $P = \{p_1, p_2, \ldots, p_n\}$. 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 $\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_i$, make sure each collection $\tilde{P}_i$ only contains papers written by one scientist, as shown in Fig. 1.

### 3.2 User feedback

Our proposed method acquires feedbacks from users, however, we should be aware that the original feedbacks collected from users inevitably contain 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 types according to the users providing them.

**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, their feedback is credible.

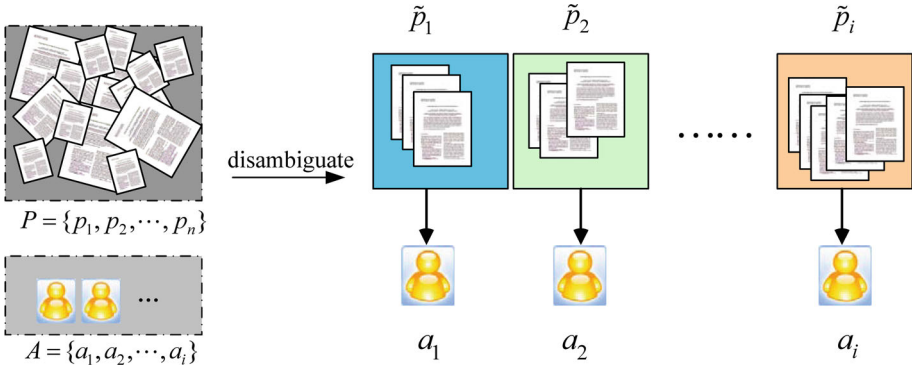


**Fig. 1** Name disambiguation in scientific cooperation network

**Table 1** Features for a pair of papers $\left(p_i^a, p_j^a\right)$

| Feature | Description |
|---|---|
| Co-author | $p_i$ and $p_j$ share at least one co-author |
| Co-Org | The org of author $a$ in $p_i$ and $p_j$ are the same |
| Citation | Paper $p_i$ cites $p_j$ in the reference, or vice versa |
| Title-Similarity | Similarity between titles of $p_i$ and $p_j$ |
| Homepage | $p_i$ and $p_j$ appear on the same homepage |
| Digital-Lib | $p_i$ and $p_j$ are published at the same digital library |
| PDF-File | The org of author $a$ in $p_j$ appears in the PDF file of $p_i$, or vice versa |

**2. Credible user feedback** The feedback is provided by the friends of the ambiguous authors. The ambiguous authors' friends can be considered individuals who have co-worked with the ambiguous authors. These users are not directly involved in the creation of the target paper, they are not very sure about who are the authors, there might exists 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 traditional machine learning algorithms are generally static. 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).

$$[(p_{i1}, p_{j1}, v_n, c_n), (p_{i2}, p_{j2}, v_n, c_n), \ldots, (p_{in}, p_{j_n}, v_n, c_n)]$$

It is a sequence constructed by user feedback according to the time stamp $\{1, \ldots, n\}$. Where the value of $v_n$ is 1 or 0, denotes user feedback considers 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

In the publication data set, a paper is always associated with a variety of attributes (Lin et al. 2010). our algorithm utilizes the following features of a paper pair for name disambiguation: Co-Author, Co-Org, Citation, Title-Similarity, Homepage, Digital-Lib and PDF-File, as summarized in Table 1, and More detailed descriptions about the features can refer to Lin et al. (2010).

Besides the seven features shown in Table 1, 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.

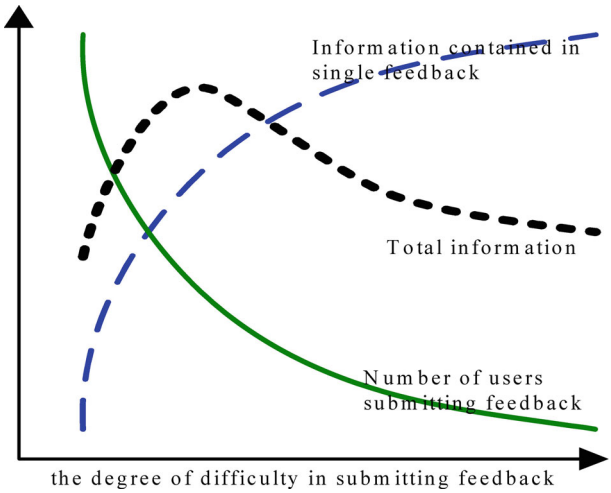


**Fig. 2** Relation between degree of difficulty in submitting user feedback and user feedback information

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 (1).

$$r_8 = \begin{cases} \frac{m}{m+n} & \text{if } m+n \neq 0 \\ 0.5 & \text{else} \end{cases} \tag{1}$$

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 (1), 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 (Fig. 2).

## 4 User feedback design

In the Internet era, it is quite popular to collect user feedback information, including explicit ways and implicit ways. Explicit user feedback forms are varied in multiple ways, for example, Wikipedia directly makes the entries editable and let users to edit them, and Google map encourages users to mark the buildings and roads. Google search engine records the users' clicking on a query and improves the search engine ranking algorithm, which is a implicit way of collecting user feedback. The advantage of the explicit way is that the collected feedback information is direct and effective, however, it requires users to edit large section of content. The advantage of implicit ways is simple and easy, however, feedback information collected in such way is not rich and the useful information always hides deeply. As shown in Fig. 3, the information of a single user feedback is proportional to the degree of difficulty in submitting the feedback, and the quantity of user feedback is inversely proportional to the

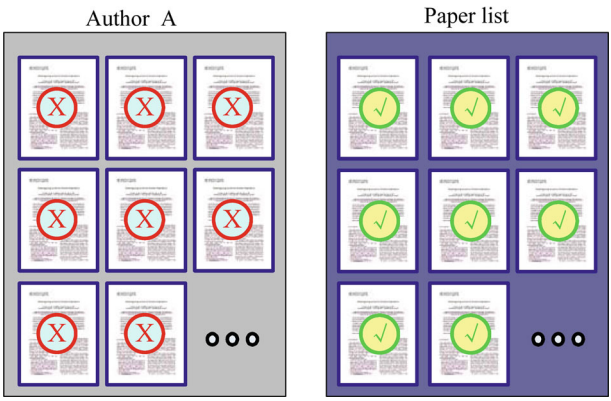


**Fig. 3** User feedback form

difficulty in submitting the feedback. So we should choose a suitable form to get the most useful user feedback information.

The user feedback form we designed for the name distinguishing method is shown in Fig. 3. When users are browsing the academic papers published by their interested scientists, namely the left of the figure, and find out some papers are not published by the author, the users can click to delete the papers. And if users find out papers on the right of the figure are actually published by the author, they can click to add the papers. The user feedback form only needs a few clicking operations and contains rich information. Besides, we must have a reasonable and unified form to record these user feedback information.

If users feedback the information of the name disambiguation result of author $A$, and $A$ is one of the disambiguating authors. The disambiguation method considers the papers published by $A$ as $\{p_{A0}, p_{A1}, \ldots, p_{An}\}$, and the papers authored by other disambiguated names as $\{p_{O0}, p_{O1}, \ldots, p_{Om}\}$.

If a user regards one academic paper $p_{Aj}$ of author $A$ in the disambiguation result is not authored by $A$, then he will click the delete button on the left side of Fig. 3, this operation will create a piece of user feedback of deleting academic paper $p_{Aj}$, we record it as

$$[(p_{A0}, p_{Aj}, 0), (p_{A1}, p_{Aj}, 0) \ldots (p_{Ak}, p_{Aj}, 0) \ldots (p_{Am}, p_{Aj}, 0)]; username;time$$

where $(p_{Ak}, p_{Aj}, 0)$ is a user feedback unit, which represents $p_{Ak}$ and $p_{Aj}$ are not authored by the same person. $username$ records the information of the user submitting the feedback, and $time$ represents the time when the user submitting the feedback.

If the user regards some academic paper $p_{Oj}$ of the authors with disambiguated name is actually authored by $A$, then he can click the add button of the right side of the Fig. 3 to submit a piece of feedback of adding academic paper $p_{Oj}$, we record it as

$$[(p_{A0}, p_{Oi}, 1), (p_{A1}, p_{Oj}, 1) \ldots (p_{Ak}, p_{Oj}, 1) \ldots (p_{Am}, p_{Oj}, 1)]; username;time$$

where $(p_{Ak}, p_{Oj}, 1)$ donates academic papers $p_{Ak}$ and $p_{Oj}$ are authored by the same person.

## 5 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

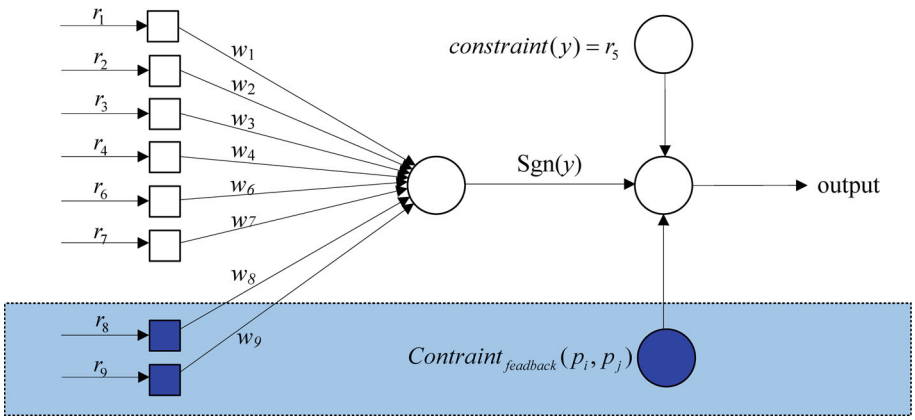


**Fig. 4** Incorporating user feedback into constraint-based perceptron

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 (2).

$$Constraint_{feedback}(p_i, p_j) = \begin{cases} 1 & p_i \text{ and } p_j \text{ belong to one author} \\ -1 & p_i \text{ and } p_j \text{ belong to different authors} \end{cases} \tag{2}$$

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 (3).

$$Output(p_i, p_j) = \begin{cases} 1 & c(y) = 1 \text{ or } c(y) = 0 \text{ and } Sgn(y) = 1 \\ 0 & c(y) = -1 \text{ or } c(y) = 0 \text{ and } Sgn(y) = 0 \end{cases} \tag{3}$$

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 (4).

$$Sgn(y) = \begin{cases} 1 & \text{if } \omega * y + b > 0 \\ 0 & \text{else} \end{cases} \tag{4}$$

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 Fig. 4. 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.

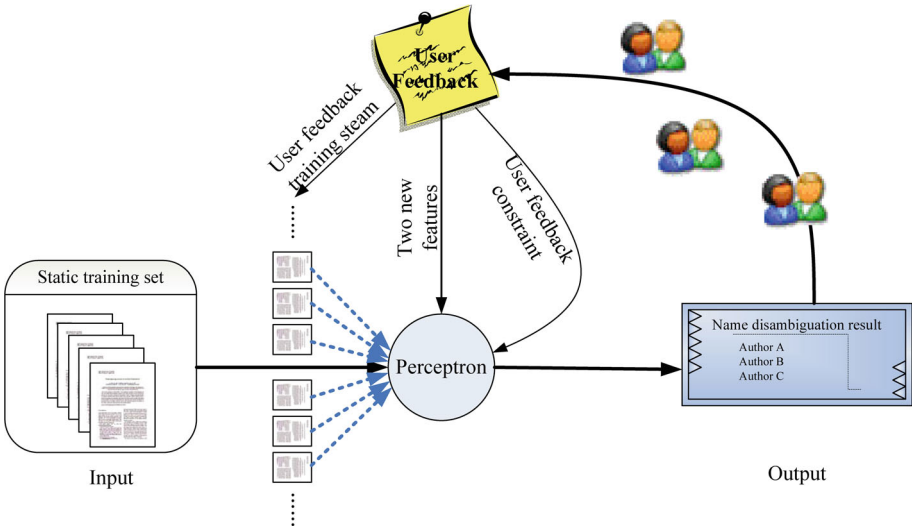


**Fig. 5** Procedures of our proposed method

The procedures of our proposed method is shown in Fig. 5. 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.

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

$$[(p_{i1}, p_{j1}, v_1, c_1), (p_{i2}, p_{j2}, v_2, c_2), \ldots, (p_{in}, p_{jn}, v_n, c_n)]$$

**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 $\alpha_f$ is the learning rate when using user feedback as training set, the value of $\alpha_f$ 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 $\alpha_f$ should be greater than the value of $\alpha_f$ sample when the value of $c_n$ in credible sample is 0.

As seen from Fig. 5, our approach is differ from previous methods without introducing user interactions that it can correct itself, since it has continuous learning ability.

## 6 Experiment

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

### 6.1 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 41 real person names with their 2,638 papers. For these names, some only have a few persons. For example, "Juan Carlos Lopez" only represents one person, and "Robert Schreiber", "Thomas Tran", "Shu lin" and "Daniel Massey" represent two. However, there are 25 different "Bing Liu" and 24 different persons named "Michael Smith". Table 2 will give some detail information about the dataset.

### 6.2 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

$$\text{Pairwise\_Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{5}$$

$$\text{Pairwise\_Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{6}$$

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

### 6.3 Experiment design

**Contribution of each single feature to the disambiguation algorithm** We adopt seven paper-pair features in our algorithm, including Co-Author, Co-Org, Citation, Title-Similarity, Homepage, Digital-Lib and PDF-File, each of which can be used as classifier solely. When the value of the feature is 1, then the output of the classifier is that the two papers are authored by the same author, otherwise, the output is that the two papers are not authored by the same author. We evaluate the effectiveness of each single feature.

**Our proposed algorithm importing different types of user feedback** We select a set of weights as the initial weights, and artificially simulate three kinds of user feedback. The feedback is provided for the perceptron to learn dynamically and dynamically adjust the weight. For creating the user feedback needs a lot of work, we only simulate 620 pieces of user feedback, of which there are 20 pieces of fully credible user feedback, the accuracy is 100%, there are 100 pieces of credible user feedback, the accuracy is 80% and there are 500 pieces of generally credible user feedback, the accuracy is 60%.

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

| Name | Pub | Person |
|---|---|---|
| Michael Smith | 38 | 24 |
| Yoshio Tanaka | 43 | 3 |
| Hui Yu | 32 | 22 |
| Ping Zhou | 36 | 18 |
| Manuel Silva | 74 | 7 |
| Eric Martin | 85 | 5 |
| Fei Su | 40 | 4 |
| Hiroshi Tanaka | 43 | 8 |
| Lei Jin | 20 | 8 |
| Thomas Wolf | 36 | 9 |
| Thomas Tran | 16 | 2 |
| Yun Wang | 57 | 22 |
| Gang Luo | 47 | 9 |
| R. Ramesh | 46 | 9 |
| Feng Pan | 73 | 15 |
| Thomas D. Taylor | 4 | 3 |
| Jim Gray | 200 | 9 |
| Sanjay Jain | 217 | 5 |
| Shu lin | 76 | 2 |
| Daniel Massey | 43 | 2 |
| David C. Wilson | 65 | 5 |
| Philip J. Smith | 33 | 3 |
| Yang Yu | 72 | 20 |
| Qiang shen | 70 | 3 |
| Michael Lang | 24 | 6 |
| Charles Smith | 7 | 4 |
| Kai Zhang | 66 | 24 |
| Robert Schreiber | 59 | 2 |
| Satoshi Kobayashi | 38 | 6 |
| David Jensen | 53 | 4 |
| Koichi Furukawa | 77 | 3 |
| Thomas Hermann | 47 | 9 |
| Cheng Chang | 27 | 5 |
| Bing Liu | 215 | 25 |
| David E. Goldberg | 231 | 3 |
| Rakesh Kumar | 96 | 12 |
| Richard Taylor | 35 | 16 |
| Juan Carlos Lopez | 36 | 1 |
| Ajay Gupta | 36 | 9 |
| Michael Siegel | 54 | 6 |
| Michael Wagner | 71 | 15 |

**Contribution of each single type of user feedback to the proposed algorithm** In our proposed algorithm, we divide the user feedback into three types, that is fully credible user feedback, credible user feedback and generally credible user feedback. Each type of user feedback can be used to the proposed algorithm to revise the disambiguated results and improve the result. Here we evaluate the effectiveness of each type of user feedback.

**Baseline methods** For comparison with our approach, we employ the SA-Cluster algorithm and Pairwise-Classification algorithm. The SA-Cluster is a graph clustering (Zhou et al. 2009) with the coauthor relationship as the edge and all the other relationships as the attribute features. The Pairwise-Classification (Lin et al. 2010) which combines paper-pair features to the constraint-based perceptron is used as the no-feedback method for comparison with our approach.

### 6.4 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 outperforms all the baselines. Our proposal outperforms the method without importing user feedback in Pairwise_Precision, Pairwise_Recall and Pairwise_F-measure.

Figure 6 shows the contributions of each single feature. Co-Author has the highest Pairwise F-measure because the author names in each paper are complete, compared to the other features. The Recall of Co-Org is also very low because the organization information is very incomplete. However, the precision is very high because at the same organization two authors merely have the same name. 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. Title Similarity gives very good performance. Because the title information is complete and an author usually publishes a serious of papers in one direction. Homepage has the third highest F-measure and the precision is 100%. Since the homepage normally contains only the owner's papers only. Digital-Lib supplements Citation data while PDF-File supplements Co-Org information. The information of PDF-File is difficult to be fetched, so the F-measure is low. More detailed discussions about the contribution of each single feature can refer to our previous work (Lin et al. 2010).

Specifically, as seen from Fig. 6, our proposal 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 Pairwise_F-measure compared with the no-feedback method Pairwise-Classification.

Figure 7 shows the contributions of each type of user feedback. We figure out that fully credible user feedback has the highest Pairwise_F-measure because the fully credible user 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 paper, so their feedback is surely credible. However, the recall of the fully credible user feedback is relatively low. The credible user feedback has the medium contribution for the disambiguation method, for this type of feedback is from co-workers of the ambiguous authors, they are not directly involved in the creation of the target paper. The contribution of the generally credible user feedback is almost the same as the credible user feedback, though the precision of the generally credible user feedback is low, however, it has the highest recall. Since this type of feedback is from readers of the paper, readers are willing to publish their feedback about the disambiguation results.

**Table 3** Results for 20 real names

| Name | Proposed method | | | Baseline method | | |
|---|---|---|---|---|---|---|
| | Pre. | Rec. | F | Pre. | Rec. | F |
| Robert Schreiber | 92.05 | 73.41 | 81.68 | 90.12 | 73.01 | 80.67 |
| Michael Smith | 100 | 100 | 100 | 99.4 | 99.86 | 99.63 |
| Hiroshi Tanaka | 95.56 | 87.43 | 91.31 | 94.25 | 87.04 | 90.5 |
| Satoshi Kobayashi | 89.36 | 33.33 | 48.55 | 89.02 | 31.56 | 46.6 |
| Philip J. Smith | 96.93 | 72.72 | 83.1 | 96.48 | 71.59 | 82.19 |
| David E. Goldberg | 100 | 56.04 | 71.83 | 99.89 | 55.87 | 71.66 |
| Yoshio Tanaka | 100 | 71.3 | 83.25 | 98.89 | 70.16 | 82.08 |
| Hui Yu | 100 | 67.65 | 80.7 | 99.94 | 65.1 | 78.84 |
| Feng Pan | 100 | 83.95 | 91.28 | 98.2 | 80.96 | 88.75 |
| Qiang shen | 98.41 | 100 | 99.2 | 98.03 | 97.16 | 97.59 |
| Lei Jin | 88.99 | 93.88 | 91.36 | 88.2 | 92.46 | 90.3 |
| Yang Yu | 100 | 68.12 | 81.04 | 99.53 | 67.46 | 80.41 |
| Ping Zhou | 99.12 | 98.26 | 98.69 | 99.08 | 98.08 | 98.6 |
| Rakesh Kumar | 100 | 97.49 | 98.73 | 100 | 96.43 | 98.18 |
| David Jensen | 100 | 100 | 100 | 99.72 | 100 | 99.86 |
| Thomas Wolf | 100 | 67.82 | 80.82 | 99.91 | 66.76 | 80.04 |
| Michael Lang | 93.76 | 85.72 | 89.24 | 91.5 | 84.03 | 87.61 |
| Thomas D. Taylor | 100 | 89.05 | 94.21 | 99.91 | 87.6 | 93.35 |
| Manuel Silva | 100 | 97.74 | 98.86 | 99.4 | 94.16 | 96.71 |
| Charles Smith | 100 | 63.03 | 77.32 | 99.75 | 60.72 | 75.49 |
| Koichi Furukawa | 93.55 | 90.48 | 91.99 | 93.32 | 83.23 | 90.15 |
| Thomas Tran | 72.72 | 96.93 | 83.1 | 69.23 | 97.03 | 78.16 |
| Thomas Hermann | 56.04 | 100 | 71.83 | 61.46 | 98.26 | 67.79 |
| Richard Taylor | 71.3 | 100 | 83.25 | 70.15 | 98.67 | 80.64 |
| Jim Gray | 67.82 | 99.64 | 80.82 | 65.23 | 99.01 | 77.63 |
| Juan Carlos Lopez | 85.72 | 93.76 | 89.24 | 83.72 | 90.17 | 85.61 |
| Sanjay Jain | 97.74 | 100 | 98.86 | 96.47 | 97.75 | 96.68 |
| Ajay Gupta | 63.03 | 99.5 | 77.32 | 57.71 | 97.8 | 72.72 |
| Shu lin | 79.93 | 100 | 88.84 | 77.17 | 99.51 | 86.52 |
| Michael Siegel | 92.87 | 100 | 96.3 | 92.56 | 98.89 | 94.86 |
| Eric Martin | 98.31 | 99.1 | 99.15 | 97.73 | 98.62 | 97.73 |
| Yun Wang | 67.65 | 100 | 80.7 | 65.23 | 100 | 77.88 |
| Kai Zhang | 82.55 | 100 | 90.44 | 80.66 | 99.93 | 88.47 |
| Cheng Chang | 83.95 | 100 | 91.28 | 80.64 | 99.23 | 88.25 |
| Daniel Massey | 95.24 | 100 | 97.56 | 94.26 | 100 | 96.32 |
| Fei Su | 100 | 100 | 100 | 99.5 | 99.87 | 99.71 |
| Michael Wagner | 84.54 | 93.97 | 89.01 | 83.6 | 89.01 | 84.59 |
| David C. Wilson | 89.5 | 100 | 94.46 | 87.31 | 99.24 | 93.44 |
| Gang Luo | 100 | 98.41 | 99.2 | 97.15 | 97.64 | 98.32 |
| Bing Liu | 93.88 | 88.99 | 91.36 | 93.04 | 88.47 | 86.83 |
| R. Ramesh | 68.12 | 100 | 81.04 | 64.55 | 97.38 | 77.86 |

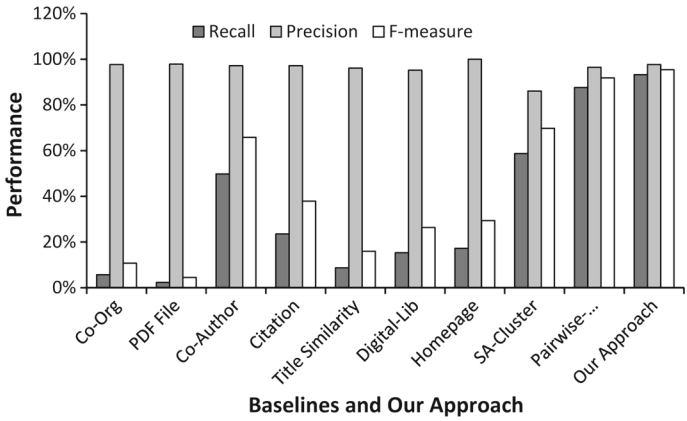


**Fig. 6** Comparison of all the baselines and our approach

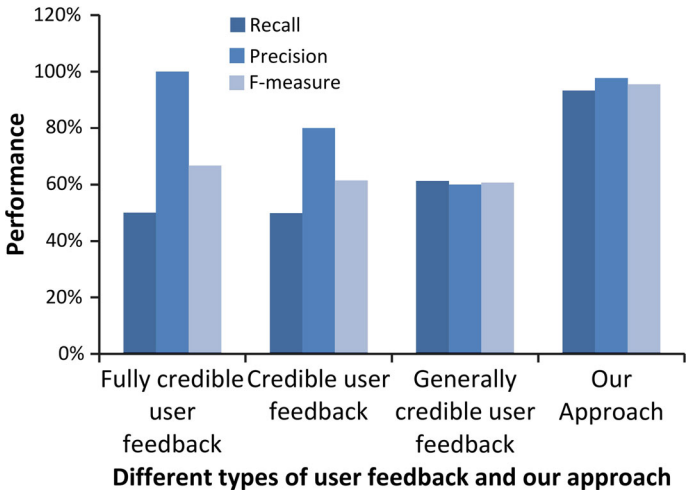


**Fig. 7** The effectiveness of different types of user feedback

## 7 Conclusion

This paper focuses on the problem of name disambiguation in scientific network. We have proposed a constraint-based perceptron to handle the problem. The method can corporate features extracted from scientific cooperation network and user feedback to the model. The method combines 7 different paper-pair features: Co-Author, Co-Org, Citation, Title-Similarity, Homepage, Digital-Lib, and PDF-File. Furthermore, new features extraced from user feedback is imported into the perceptron, which can revise the perceptron with new constraints and continuously train the perceptron. Despite the noise of user feedback, bringing in user feedback gives the best result. In addition, we evaluate the effect of different types of user feedback to name disambiguation. It has been proved that fully credible user feedback has the highest precision and generally credible user feedback also plays an important role in name disambiguating as it has highest recall.

In the future work, we plan to enhance the approach int the following directions: First, we will exploit more useful features, and combine all the features in a more effective manner. Second, we will design more efficient and convenient user feedback forms to attract more users to publish their feedback. Ideally, we should provide a system which can actively select only a few potentially wrong but most useful disambiguation results to query the users instead of passively waiting for users to publish their feedback. Besides, we should evaluate our method on different data sets except publication data set, such as web page data set and news page data set. Furthermore, we will consider the case that there is mis-ordering or missing authors in our method. 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.

## References

Bekkerman R, McCallum A (2005) Disambiguating web appearances of people in a social network. In: WWW, pp 463–470

Cao H, Jiang D, Pei J, He Q, Liao Z, Chen E, Li H (2008) Context-aware query suggestion by mining click-through and session data. In: KDD, pp 875–883

Chai X, Vuong BQ, Doan A, Naughton JF (2009) Efficiently incorporating user feedback into information extraction and integration programs. In: SIGMOD conference, pp 87–100

Chen Z, Kalashnikov DV, Mehrotra S (2007) Adaptive graphical approach to entity resolution. In: JCDL, pp 204–213

Cota RG, Ferreira AA, Nascimento C, Gonçalves MA, Laender AHF (2010) An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. JASIST 61(9):1853–1870

D'Angelo CA, Giuffrida C, Abramo G (2011) A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments. JASIST 62(2):257–269

Davis PT, Elson DK, Klavans J (2003) Methods for precise named entity matching in digital collections. In: JCDL, pp 125–127

Ferreira AA, Veloso A, Gonçalves MA, Laender AHF (2010) Effective self-training author name disambiguation in scholarly digital libraries. In: JCDL, pp 39–48

H Yu WK, Hatzivassiloglou V, Wilbur J (2006) A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations. ACM TOIS 24(3):380C404

Han H, Giles CL, Zha H, Li C, Tsioutsiouliklis K (2004) Two supervised learning approaches for name disambiguation in author citations. In: JCDL, pp 296–305

Han H, Zha H, Giles CL (2005) Name disambiguation in author citations using a k-way spectral clustering method. In: JCDL, pp 334–343

Jin H, Huang L, Yuan P (2009) Name disambiguation using semantic association clustering. In: ICEBE, pp 42–48

Joachims T, Granka LA, Pan B, Hembrooke H, Gay G (2005) Accurately interpreting clickthrough data as implicit feedback. In: SIGIR, pp 154–161

Kang IS, Na SH, Lee S, Jung H, Kim P, Sung WK, Lee JH (2009) On co-authorship for author disambiguation. Inf Process Manag 45(1):84–97

Lin Q, Wang B, Du Y, Wang X, Li Y (2010) Disambiguating authors by pairwise classification. Tsinghua Sci Technol 15(6):668–677

Liu Y, Zhang M, Ru L, Ma S (2006) Automatic query type identification based on click through information. In: AIRS, pp 593–600

McRae-Spencer DM, Shadbolt NR (2006) Also by the same author: Aktiveauthor, a citation graph approach to name disambiguation. In: JCDL, pp 53–54

Minkov E, Cohen WW, Ng AY (2006) Contextual search and name disambiguation in email using graphs. In: SIGIR, pp 27–34

Nguyen HT, Cao TH (2010) Exploring wikipedia and text features for named entity disambiguation. In: ACIIDS, vol 2, pp 11–20

Richardson M, Dominowska E, Ragno R (2007) Predicting clicks: estimating the click-through rate for new ads. In: WWW, pp 521–530

Tan YF, Kan MY, Lee D (2006) Search engine driven author disambiguation. In: JCDL, pp 314–315

Treeratpituk P, Giles CL (2009) Disambiguating authors in academic publications using random forests. In: JCDL, pp 39–48

Wang C, Han J, Jia Y, Tang J, Zhang D, Yu Y, Guo J (2010a) Mining advisor-advisee relationships from research publication networks. In: KDD, pp 203–212

Wang F, Tang J, Li J, Wang K (2010) A constraint-based topic modeling approach for name disambiguation. Front Comput Sci China 4(1):100–111

Whang SE, Menestrina D, Koutrika G, Theobald M, Garcia-Molina H (2009) Entity resolution with iterative blocking. In: SIGMOD, pp 219C232

Wick ML, Rohanimanesh K, Schultz K, McCallum A (2008) A unified approach for schema matching, coreference and canonicalization. In: KDD, pp 722C730

Yang KH, Peng HT, Jiang JY, Lee HM, Ho JM (2008) Author name disambiguation for citations using topic and web correlation. In: ECDL, pp 185–196

Zhang D, Tang J, Li JZ, Wang K (2007) A constraint-based probabilistic framework for name disambiguation. In: CIKM, pp 1019–1022

Zhou Y, Cheng H, Yu JX (2009) Graph clustering based on structural/attribute similarities. PVLDB 2(1):718–729