Improving Author Name Disambiguation with User Relevance Feedback

Anderson A. Ferreira[1,2], Tales Mota Machado[1]
, Marcos André Gonçalves[2]

[1] Departamento de Computação, Universidade Federal de Ouro Preto
[2] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais
ferreira@dcc.ufmg.br; talesmmachado@gmail.com; mgoncalv@dcc.ufmg.br

Abstract. Author name ambiguity in the context of bibliographic citations is a very hard problem. It occurs when there are citation records of a same author under distinct names or when there exists citation records belonging to distinct authors with very similar names. Among the several methods proposed in the literature, the most effective ones are those that perform a direct assignment of the records to their respective authors by means of the application of supervised machine learning techniques. However, those methods usually need large amounts of labeled training examples to properly disambiguate the author names. To deal with this issue, in previous work, we have proposed a method that automatically obtains and labels the training examples, showing competitive performance compared to representative author name disambiguation methods. In this work, we propose to improve our previous method by exploiting user relevance feedback. In more details we select a very small portion of the citation records for which our method was mostly unsure about the correct authorship and ask the administrators for labeling them. This feedback is then used to improve the effectiveness of the whole process. In our experimental evaluation, we observed that with a very small labeling effort (usually around 5% of the records), the overall disambiguation effectiveness improves by almost 10% on average, with gains of up to 61% in some of the largest ambiguous groups.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

Keywords: Bibliographic Citation, Digital Library, Name Disambiguation, Relevance Feedback

1. INTRODUCTION

Author name ambiguity in the context of bibliographic citations is one of the hardest problem currently faced by the digital library (DL) community. This problem occurs when a set of citation records[1] contains ambiguous author names, i.e., the same author may appear under distinct names, or distinct authors may have very similar names. This problem decreases the quality and reliability on the information that can be obtained from the digital library repositories as well as the quality of services such as searching and browsing that rely on this information.

To illustrate the problem, Table 1 shows a set of three citation records $\{c_1, c_2, c_3\}$ so that each record has its author names identified by r_j, $1 \leq j \leq 16$. For each record c_i, each name r_j is a reference to an author and has a list of attributes associated with it, such as, coauthor names (i.e., the list of references to other authors of the same citation record), work title, publication venue title, publication year and

[1]Here understood as a set of bibliographic attributes, such as author names, work title and publication venue title, of a particular publication.

This research is partially funded by the InWeb - The National Institute of Science and Technology for the Web (MCT/CNPq/FAPEMIG grant number 573871/2008-6) and by the authors's individual scholarships and research grants from CAPES, CNPq, and FAPEMIG.

Table I. Illustrative example

Citation Id	Citation
c_1	(r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma. Building re-usable dictionary repositories for real-world text mining. CIKM, 2010.
c_2	(r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English. Enabling analysts in managed services for CRM analytics. KDD, 2009.
c_3	(r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. HSCC, 2010.

so on. Examining Table 1, we see examples of synonyms and polysems, which are subproblems of the name ambiguity problem. Author names r_3 and r_{15} are examples of polysems where r_3 refers to "Ajay Gupta" from IBM Research India and r_{15} refers to "Aarti Gupta" from NEC Laboratories America, USA. Author names r_3 and r_7 are examples of synonyms.

To help solving such a problem in a scalable and automated manner, an automatic *author name disambiguation* procedure may be applied to the digital library content. More formally, the author name disambiguation procedure may be formulated as follows. Let $C = \{c_1, c_2, ..., c_k\}$ be a collection of *citation records*. Each citation record c_i has a list of *attributes* which includes at least author names, work title and publication venue title. With each attribute in a citation is associated a specific value, which may be composed of several *elements*. In case of the attribute "author names", an element corresponds to the name of a single unique author. Each author name element is a *reference* r_j to an author. In case of the other attributes, an element corresponds to a word/term. The objective of the disambiguation task is to produce a disambiguation function that is used to partition the set of references to authors $\{r_1, r_2, \ldots, r_m\}$ into n sets $\{a_1, a_2, \ldots, a_n\}$, so that each partition a_i contains ideally all the references to a same author and no references to other authors. We consider that each reference r to an author extracted from a citation record c contains at least the following attributes: author name, coauthor names that contain the other references extracted from the record c, work title and publication venue title of the citation c.

To disambiguate the bibliographic citations of a digital library (DL), first we may split the set of references to authors into groups of references whose values of the author name attribute are ambiguous. These are called *ambiguous groups* (i.e., groups of references having the value of the author name attribute with similar names). The ambiguous groups may be obtained by using blocking methods [On et al. 2005] which address scalability issues avoiding the need for comparisons among all references.

The challenges of dealing with name ambiguity in citation records have led to the proposal of a large number of methods [Bhattacharya and Getoor 2007; Cota et al. 2010; Carvalho et al. 2011; Culotta et al. 2007; Fan et al. 2011; Ferreira et al. 2010; Ferreira et al. 012b; Han et al. 2004; Han et al. 2005; Huang et al. 2006; Kanani et al. 2007; Kang et al. 2009; Levin and Heuser 2010; On et al. 2006; Pereira et al. 2009; Shu et al. 2009; Soler 2007; Song et al. 2007; Tang et al. 2012; Torvik et al. 2005; Treeratpituk and Giles 2009; Veloso et al. 2012; Yang et al. 2008]. One such a challenge is that, usually, only a minimum set of attributes is available to work with (in most case only author names and publication and venue titles) which may not be enough to correctly disambiguate all the references of a DL.

The author name disambiguation methods have been generally classified as performing either [Ferreira et al. 012a]: *author grouping*, which includes methods that attempt to group references to the same author using some type of similarity between them; or *author assignment*, including methods that try to directly assign the references to their respective authors.

Among the several methods proposed in the literature, the most effective ones seem being *author*

assignment methods that exploit supervised machine learning techniques [Ferreira et al. 012a]. However, those methods usually need large amounts of labeled training examples to properly disambiguate the author names, which are costly and laborious to obtain. Moreover, changes in publication patterns may require that these training sets may be periodically rebuild in order to capture these new patterns. To deal with those issues, in previous work [Ferreira et al. 2010], we have proposed a method that automatically obtains and labels the training examples, showing competitive performance compared to representative author name disambiguation methods. In this work, we improve upon our previous proposal by exploiting user relevance feedback. In more details we select a very small portion of the set of references for which our method was most unsure about the correct authorship and ask the administrators for labeling them. These labeled references are then incorporated into the training data automatically selected by our self-training author name method. In our experimental evaluation, we observe that with a very small labeling effort (usually around 5% of all DL records), the disambiguation effectiveness improves by almost 10% on average in a collection under the pairwise F1 metric, with gains of up to 61% on some of the largest ambiguous groups.

This article is organized as follows. Section 2 discusses related work. Section 3 describes the self-training author name disambiguation method we explore. Section 4 describes our strategy to extend this disambiguation method with relevance feedback. Section 5 presents the results of our experimental evaluation. Finally, Section 6 presents our conclusions and offers possible directions for future work.

2. RELATED WORK

2.1 Author Name Disambiguation

Most of the automatic name disambiguation methods proposed in the literature adopt solutions that may be classified according to the main type of approach that exploit. Ferreira et al. [2012a] broadly classify these approaches as performing either: *author grouping* [Han et al. 2005; Torvik et al. 2005; Huang et al. 2006; On et al. 2006; Bhattacharya and Getoor 2007; Culotta et al. 2007; Kanani et al. 2007; Kang et al. 2009; On and Lee 2007; Soler 2007; Song et al. 2007; Yang et al. 2008; Pereira et al. 2009; Torvik and Smalheiser 2009; Treeratpituk and Giles 2009; Cota et al. 2010; Ferreira et al. 2010; Fan et al. 2011; Carvalho et al. 2011], encompassing methods that try to group the references to the same author using some type of similarity among reference attributes; or *author assignment* [Han et al. 2004; Han et al. 2005; Bhattacharya and Getoor 2006; Ferreira et al. 2010; Tang et al. 2012; Veloso et al. 2012; Ferreira et al. 012b], covering methods that aim at directly assigning the references to their respective authors.

Author grouping methods attempt to group references to a same author using some clustering technique along with a similarity function. The similarity function is applied to the attributes of the references (or group of references) in order to decide whether to place references in a same group or not, using the clustering technique. This similarity function may be predefined (based on existing ones and depending on the type of the attribute) [Han et al. 2005; Bhattacharya and Getoor 2007; On and Lee 2007; Soler 2007; Cota et al. 2010; Carvalho et al. 2011], learned using a supervised machine learning technique [Torvik et al. 2005; Huang et al. 2006; Culotta et al. 2007; Torvik and Smalheiser 2009; Treeratpituk and Giles 2009; Levin et al. 2012], or extracted from the relationships among authors and coauthors, usually represented as a graph [On et al. 2006; Levin and Heuser 2010; Fan et al. 2011].

Author assignment methods attempt to directly assign the references to their corresponding author using either a supervised classification technique [Han et al. 2004; Ferreira et al. 2010; Veloso et al. 2012; Ferreira et al. 012b] or a model-based clustering technique [Han et al. 2005; Bhattacharya and Getoor 2006; Tang et al. 2012]. These methods infer models that represent the authors (for instance, the probabilities of an author publishing an article with other (co-)authors, in a given publication venue and using a list of specific terms in the work title).

In this work, we extend a self-training author assignment disambiguation method by exploiting user relevance feedback. In more details, we select a very small portion of the references for which our method was not sure about the correct authorship and ask the administrators to label them. These labeled references are inserted into the training data automatically selected by the self-training author name disambiguator [Ferreira et al. 2010] that infers a function to disambiguate the remaining references.

2.2 Relevance Feedback

Relevance Feedback has been largely used mainly in search tasks in Information Retrieval [Baeza-Yates and Ribeiro-Neto 2008]. In such task, the results obtained from a given search are evaluated as relevant or not by the user who issued the query. This information about the relevance of the retrieved documents is given as feedback to the system which modifies the original query with terms belonging to the indicated documents. The new modified query is then used to retrieve new documents and this process continues until the user is satisfied or gives up.

Our idea of applying the user feedback for improving the disambiguation process is quite similar, with the difference that we focus on the cases in which the method is most uncertain about the correct authorship. We discuss how we measure this uncertainty in Section 3.2.

To the best of our knowledge, in the author name disambiguation task, the only other work to explore some type of user feedback is ADANA (standing for Active Name Disambiguation) [Wang et al. 2011]. ADANA works in an interactive mode actively choosing only a few potentially erroneous disambiguated results, after a disambiguation process has been run, in order to ask the user for corrections, instead of passively waiting for user inputs. The active selection aims to minimize the number of interactions needed to maximize effectiveness. Notice that in order to obtain good results, the authors make use of a lot of additional information such as affiliation, bibliographic references, etc., that is usually not available in most cases. Differently, we here focus on the most common case when we have only the minimum amount of information in a citation (i.e., author and coauthor names, publication and venue titles). Another important differences are that ADANA can be considered as an author grouping method, while ours lies in the author assignment class of methods and that the users in ADANA are required to "correct" potentially incorrectly made decisions while we defer this decision to the end of the process and exploit the relevance feedback in some uncertain cases to help to solving the remaining ones.

3. SELF-TRAINING ASSOCIATIVE AUTHOR NAME DISAMBIGUATION

In this section, we describe the Self-training Author Name Disambiguation method (SAND) that will be extended with relevance feedback. SAND follows a two-step approach as described in [Ferreira et al. 2010; Ferreira 2012]. These steps are applied after a well-known pre-processing procedure, which includes blocking, stop-word removal, and stemming[2]. In the following subsections we will present the SAND steps.

3.1 The Unsupervised Step

The unsupervised step aims to automatically produce training examples for the second step (the supervised one). To obtain these examples, first we organize references within each ambiguous group into clusters, so that each cluster contains (ideally) references to a same author. The key intuition is

[2]Stop-word removal and stemming are performed on the words that compose work and publication venue titles. Then, authors with ambiguous names are grouped together (i.e., blocked), so that ambiguous groups are created. Disambiguation operations are performed within each ambiguous group, so that useless comparisons involving non-ambiguous authors are avoided.

to associate each cluster with an author label, so that references within each cluster can be exploited as training examples.

In order to work properly as training examples, the extracted clusters must be as pure as possible, in the sense that each cluster must be likely to contain only references that are associated with the same author. A simple way to extract pure clusters is to ensure that each cluster contains only one reference. In this case, clusters are totally pure, however, totally fragmented (that is, references associated with the same author are placed in different clusters). Fragmented clusters are detrimental for training, since authorship references associated with the same author would receive different author labels. So, our strategy to automatically produce examples is to extract pure clusters, and then discard those clusters that increase fragmentation. This strategy is discussed next.

Extracting Pure Clusters. To extract pure clusters, we exploit highly discriminative attributes aiming to place in different clusters references associated with different authors. More specifically, pure clusters are extracted by exploiting recurring patterns in the coauthorship graph, that is, two authorship references are placed together in the same cluster if both references have at least one coauthor in common. We have based this strategy on the general observation that only very rarely two ambiguous authors share a coauthor [Cota et al. 2007]. While simple, this strategy tends to extract highly pure clusters. Unfortunately, this strategy also tends to fragment references associated with the same author into multiple clusters. This is expected, since some authors are likely to have many different coauthors and these coauthors may not act as coauthors among themselves.

Discarding Fragmented Clusters. In order to produce the best training possible set, which ideally contains only one cluster per author in the collection, we need to discard fragmented clusters, i.e., two or more clusters with references to a same author. The process of identification of fragmented clusters starts by sorting clusters in descending order of size (i.e., the number of references in the cluster). The result is a sorted list $\mathcal{C}$ of clusters. The identification process continues and, at the first iteration, the largest cluster in $\mathcal{C}$ is inserted into the set of selected clusters, denoted as $\mathcal{S}$. Also, the selected cluster is removed from $\mathcal{C}$. The next clusters in $\mathcal{C}$ to be inserted into $\mathcal{S}$ should be as dissimilar as possible to the clusters already in $\mathcal{S}$. The key intuition is that candidate clusters in $\mathcal{C}$ that are dissimilar to clusters in $\mathcal{S}$ are those more likely to contain references associated with authors not in $\mathcal{S}$.

In order to compute the similarity among clusters, we employed the well-known cosine function [Baeza-Yates and Ribeiro-Neto 2008]. Specifically, each reference is represented as a feature vector, where each coauthor name and each word of work title and publication venue title is a feature, and the similarity is calculated using the centroid of the vectors (i.e., references) in the clusters. The identification of fragmented clusters continues by evaluating the next candidate cluster $c \in \mathcal{C}$ to be inserted into $\mathcal{S}$. Candidate $c \in \mathcal{C}$ is inserted into $\mathcal{S}$ if:

$$\forall s \in \mathcal{S}, \phi(c, s) \leq \phi_{max} \tag{1}$$

That is, c is inserted into $\mathcal{S}$ if the similarity value $\phi(c, s)$ between c and each cluster $s \in \mathcal{S}$ is at most ϕ_{max} (which is a user-specified threshold)[3]. Then, c is removed from $\mathcal{C}$ and the iteration continues with the next candidate cluster in $\mathcal{C}$. The process finally stops when there is no more candidate clusters, and $\mathcal{C}$ is finally empty. In the end, references in each cluster $s \in \mathcal{S}$ are inserted into the training data $\mathcal{D}$. Each reference receives the author label of the corresponding cluster. The remaining references that were not included in $\mathcal{D}$ (i.e., references associated with fragmented clusters that were not included in $\mathcal{S}$) compose the test set $\mathcal{T}$.

[3] $\phi(c, s)$ is a function that measures the similarity between clusters c and s

3.2 The Supervised Step

The unsupervised step produces a set of training examples $\mathcal{D}$, which consists of a set of records of the form <r,a>, where r is a reference to an author (represented as a list of m feature-values or $\{f_1, f_2, \ldots, f_m\}$) and a is a label that identifies the correct author of r. The supervised step uses $\mathcal{D}$ to produce a disambiguation function that relates features to the correct author. Our disambiguation function is a function from $\{f_1, f_2, \ldots, f_m\}$ to $\{a_1, a_2, \ldots, a_n\}$ that is used to predict the correct author for references in the test set $\mathcal{T}$.

The supervised step is inspired by the observed fact that, frequently, there are strong associations between features $\{f_1, f_2, \ldots, f_m\}$ and specific authors $\{a_1, a_2, \ldots, a_n\}$. Such associations are uncovered from $\mathcal{D}$, producing a disambiguation function $\{f_1, f_2, \ldots, f_m\} \rightarrow \{a_1, a_2, \ldots, a_n\}$ [Veloso et al. 2006]. Typically, these associations are expressed using rules[4] of the form $\mathcal{X} \rightarrow a_1$, $\mathcal{X} \rightarrow a_2$, ..., $\mathcal{X} \rightarrow a_n$, where $\mathcal{X} \subseteq \{f_1, f_2, \ldots, f_m\}$. In the following discussion, we denote as $\mathcal{R}$ an arbitrary rule set. Similarly, we denote as $\mathcal{R}_{a_i}$ a subset of $\mathcal{R}$ that is composed of rules of the form $\mathcal{X} \rightarrow a_i$ (i.e., rules predicting author a_i). A rule $\mathcal{X} \rightarrow a_i$ is said to match a record x if $\mathcal{X} \subseteq x$ (i.e., x contains all features in $\mathcal{X}$) and this rule is included in $\mathcal{R}^x_{a_i}$. That is, $\mathcal{R}^x_{a_i}$ is composed of rules predicting author a_i and matching record x. Obviously, $\mathcal{R}^x_{a_i} \subseteq \mathcal{R}_{a_i} \subseteq \mathcal{R}$. The *test set* (referred to as $\mathcal{T}$) consists of records <r,?> for which only the feature-values of the reference r to an author are known, while the author of r is unknown. Disambiguation functions obtained from $\mathcal{D}$ are used to estimate the correct authors of such records in $\mathcal{T}$.

Prediction. To predict the author of a reference, we use a widely used statistic, called confidence [Agrawal et al. 1993] (denoted as $\theta(\mathcal{X} \rightarrow a_i)$), to measure the strength of the association between $\mathcal{X}$ and a_i. Put simple, the confidence of the rule $\mathcal{X} \rightarrow a_i$ is given by the conditional probability of a_i being the author of record x, given that $\mathcal{X} \subseteq x$.

The probability (or likelihood) of a_i being the author of record x is estimated by combining rules in $\mathcal{R}^x_{a_i}$. More specifically, $\mathcal{R}^x_{a_i}$ is interpreted as a poll, in which each rule $\mathcal{X} \rightarrow a_i \in \mathcal{R}^x_{a_i}$ is a vote given by features in $\mathcal{X}$ for author a_i. The weight of a vote $\mathcal{X} \rightarrow a_i$ depends on the strength of the association between $\mathcal{X}$ and a_i, which is given by $\theta(\mathcal{X} \rightarrow a_i)$. The process of estimating the probability of a_i being the author of record x starts by summing weighted votes for a_i and then averaging the obtained value by the total number of votes for a_i, as expressed by the score function $s(a_i, x)$ shown in Equation 2 (where $r_j \subseteq \mathcal{R}^x_{a_i}$ and $|\mathcal{R}^x_{a_i}|$ is the number of rules in $\mathcal{R}^x_{a_i}$). Thus, $s(a_i, x)$ gives the average confidence of the rules in $\mathcal{R}^x_{a_i}$ (obviously, the higher the confidence, the stronger the evidence of authorship).

$$s(a_i, x) = \frac{\sum_{j=1}^{|\mathcal{R}^x_{a_i}|} \theta(r_j)}{|\mathcal{R}^x_{a_i}|} \tag{2}$$

The estimated probability of a_i being the author of record x, denoted as $\hat{p}(a_i|x)$, is simply obtained by normalizing $s(a_i, x)$, as shown in Equation (3). A higher value of $\hat{p}(a_i|x)$ indicates a higher likelihood of a_i being the author of x. The author associated with the highest likelihood is finally predicted as the author of record x.

$$\hat{p}(a_i|x) = \frac{s(a_i, x)}{\sum_{j=1}^{n} s(a_j, x)} \tag{3}$$

[4]These rules can be efficiently extracted from $\mathcal{D}$ using the strategy proposed in [Veloso et al. 2006].

Detecting Novel Authors. To indicate that a reference belong to an unseen author, we use the lack of rules supporting any already seen author (i.e., authors that are present in some record in $\mathcal{D}$) as evidence. The number of rules that is necessary to consider an author as an already seen one is controlled by a parameter, γ_{min}. Specifically, for a record $x \in \mathcal{T}$, if the number of rules extracted from $\mathcal{D}^x$ (which is denoted as $\gamma(x)$) is smaller than γ_{min} (i.e., $\gamma(x) < \gamma_{min}$), then the author of x is considered as a new/unseen author and a new label a_k is created to identify such author. Further, this prediction is considered as a new example and included into $\mathcal{D}$. We estimate the appropriate value for γ_{min} by performing cross-validation [Geisser 1993], which is a way to predict the fit of a disambiguation function to a hypothetical validation set.

Exploiting Reliable Predictions. To increase the coverage of the training data, additional examples may be obtained from the predictions performed using the disambiguation function. In this case, reliable predictions are regarded as correct ones and, thus, they can be safely included in the training examples. Next we define the *reliability* of a prediction.

Given an arbitrary record $x \in \mathcal{T}$, and the two most likely authors for x, a_i and a_j, we denote as $\Delta(x)$ the reliability of predicting a_i, as shown in Equation 4.

$$\Delta(x) = \frac{\hat{p}(a_i|x)}{\hat{p}(a_j|x)} \tag{4}$$

The idea is to only predict a_i if $\Delta(x) \geq \Delta_{min}$, where Δ_{min} is a threshold that indicates the minimum reliability necessary to regard the corresponding prediction as correct, and, therefore, to include it into the training data $\mathcal{D}$. An appropriate value for Δ_{min} can be obtained by performing cross-validation.

Naturally, some predictions are not enough reliable for certain values of Δ_{min}. An alternative is to abstain from such doubtful predictions. As new examples are included into $\mathcal{D}$ (i.e., the reliable predictions), new evidence may be exploited, hopefully increasing the reliability of the predictions that were previously abstained. To optimize the usage of reliable predictions, we place records in a queue, so that records associated with reliable predictions are considered first. The process works as follows. Initially, records in the test set are randomly placed in the queue. If the author of the record that is located in the beginning of the queue can be reliably predicted, then the prediction is performed, the record is removed from the queue and included into $\mathcal{D}$ as a new example. Otherwise, if the prediction is not reliable, the corresponding record is simply placed in the end of the queue and will be only processed after all other records. The process continues performing more reliable predictions first, until no more reliable predictions are possible. The remaining records in the queue (for which only doubtful predictions are possible) are then processed normally but they are not inserted into the training data $\mathcal{D}$.

4. RELEVANCE FEEDBACK FOR AUTHOR NAME DISAMBIGUATION

In this section, we describe our proposed strategy for author name disambiguation that exploits user relevance feedback. Our objective is to select a very small portion of the references for which SAND is not sure about the correct authorship and asks the administrators for labeling them. These labeled references are then incorporated into the training data that was automatically produced by self-training author name method, for the application of the supervised step. We describe this procedure in more details next.

By using Equation 4, we can consider reliable a prediction of the author of a record x for which $\Delta(x) \geq \Delta_{min}$. The unreliable predictions are the candidates for labeling. In order to try to maximize the amount of useful information that can be obtained by labeling these instances with the minimum effort, we start the labeling by considering first the references with the most unreliable predictions, ordered by $\Delta(x)$. Algorithm 1 describes this process in more details.

Algorithm 1 Author Name Disambiguation with User Relevance Feedback

Input: Set of references R
Input: Threshold ϕ_{max}
Input: Percentage p to select references
Output: Set of disambiguated references $\mathcal{R}'$

1: $[\mathcal{D}, \mathcal{T}] \Leftarrow$ perform unsupervised step on the set of references R using ϕ_{max}
2: $\mathcal{U} \Leftarrow$ get doubtful predictions from test set $\mathcal{T}$ using training data $\mathcal{D}$
3: $\mathcal{U} \Leftarrow$ sort $\mathcal{U}$ in ascending order of $\Delta(x)$
4: $\mathcal{U}' \Leftarrow$ percentage p of references on top of $\mathcal{U}$
5: $\mathcal{U}' \Leftarrow$ labeling $\mathcal{U}'$
6: $\mathcal{D}' \Leftarrow \mathcal{D} \bigcup \mathcal{U}'$
7: $\mathcal{T}' \Leftarrow \mathcal{T} - \mathcal{U}'$
8: $\mathcal{R}' \Leftarrow$ perform supervised step on $\mathcal{T}'$ using $\mathcal{D}'$
9: **return** $\mathcal{R}'$

First, we select the training data $\mathcal{D}$ and test set $\mathcal{T}$ (line 1) from the set of references using the SAND unsupervised step. Next, using the supervised step (first round of the supervised step), we select the records whose predictions are doubtful (i.e., unreliable) and these records are inserted into set $\mathcal{U}$. In more detail, we select the records to be inserted into $\mathcal{U}$ by running SAND's supervised step with the training data $\mathcal{D}$ and test set $\mathcal{T}$ and extract from $\mathcal{T}$ those records whose reliability $\Delta(x)$ is smaller than Δ_{min}. In lines 3 and 4, we select the records to be labeled by the administrator. In line 3, we sort the records in $\mathcal{U}$ in ascending order of $\Delta(x)$ to have in the first positions of $\mathcal{U}'$ the records with the most doubtful predictions and, in line 4, we select a percentage of them to be labeled, what happens in line 5. Next, we make a new training data $\mathcal{D}'$ with the records from $\mathcal{D}$ and $\mathcal{U}'$ (line 6). And, finally, we remove each record $x \in \mathcal{U}'$ from $\mathcal{T}$ and perform the supervised step with these new training data $\mathcal{D}'$ and test set $\mathcal{T}'$ (second round of the supervised step).

5. EXPERIMENTAL EVALUATION

In this section we present experimental results that demonstrate the effectiveness of our proposed strategy. We first describe the collections, the baselines and the evaluation metric. Then, we discuss the effectiveness of our strategy in comparison with the baselines.

5.1 Collections

We use two collections of references derived from DBLP[5] to evaluate our strategy for disambiguating author name with user feedback relevance. These collections contain several ambiguous groups (i.e., groups of references with ambiguous author names).

The first collection, hereafter referred to as DBLP, contains references extracted from DBLP. It sums up 4,287 references associated with 220 distinct authors, which means an average of approximately 20 references per author. Small variations of this collection have been used in several other works [Ferreira et al. 2010; Ferreira et al. 012b; Han et al. 2004; Han et al. 2005; Han et al. 2005; Pereira et al. 2009; Yang et al. 2008]. Its original version was created by Han et al. [2004] and they manually labeled the references. For this, they used the author's publication home page, affiliation name, e-mail and coauthor names in a complete name format, and also sent emails to some authors to confirm their authorship. They eliminated the references for which they had insufficient information to identify the correct author. Han et al. [2004] also replaced the abbreviated publication venue titles by their complete version obtained from DBLP. We used 11 ambiguous groups extracted by Han et al. [2004]

[5]http://www.informatik.uni-trier.de/∼ley/db/

with some corrections[6].

The second collection, hereafter referred to as KISTI, was built by the Korea Institute of Science and Technology Information [Kang et al. 2011] for English homonyms author name disambiguation. The top 1000 most frequent author names from late-2007 DBLP citation records were obtained jointly with their citation records. Afterwards, for each author name in each citation record, a reference was built. To disambiguate this collection, the authors submitted a query composed of the surname of the author and the work title of each reference to the Google search engine, aiming at finding personal publication pages. The first 20 web pages retrieved for each query were manually checked to identify the correct personal publication page for each authorship record. This identified page was then used to disambiguate the record. This collection has 37,613 citation records, 881 groups of same-name persons and 6,921 authors[7]. Due to the very large number of experiments and comparisons we perform, in our evaluation we use the top 40 most frequent ambiguous groups, i.e., the groups with the largest number of ambiguous authors. These groups are potentially the most difficult ones to disambiguate. In this subset, there are 7,841 citation records, 40 groups of same-name persons and 1,132 authors, with an average of 28.3 different ambiguous authors per group (with a maximum of 59).

5.2 Baselines

We used four baselines in our experiments: SAND that was discussed in Section 3, the two supervised author assignment methods proposed by Han et al. [2004] and the supervised author grouping method proposed by Huang et al. [2006].

The first method proposed by Han et al. [2004], referred to as NB, uses a naïve Bayes model, a generative statistical model frequently used in word sense disambiguation tasks, to capture all writing patterns in the authors' citations. The second method, referred to as SVM, is based on Support Vector Machines, which are discriminative models basically used as a classifier [Mitchell 1997]. An important difference between the two techniques is that a naïve Bayes model requires only positive examples to learn about the writing patterns whereas SVMs require both positive and negative examples to learn how to identify the author. The method proposed by Huang et al. [2006], referred to as LaSVM-DBSCAN, uses a clustering strategy based on the DBSCAN method [Ester et al. 1996] to cluster the references by author after an SVM-based classifier, LaSVM [Bordes et al. 2005], is applied to create a similarity function among the records.

5.3 Evaluation Metrics

In order to evaluate the effectiveness of the proposed disambiguation method, we used two evaluation metrics: the K metric, and pairwise F1. These metrics allow us to compare different disambiguation methods under a number of different criteria, which is not usually done in the literature.

In the discussion that follows, we describe these metrics. The key idea is to compare the clusters extracted by disambiguation methods against ideal, perfect clusters, which were manually extracted. Hereafter, a cluster extracted by a disambiguation method will be referred to as *empirical cluster*, while a perfect cluster will be referred to as *theoretical cluster*.

K Metric

The K metric [Lapidot 2002] determines the trade-off between the average cluster purity (ACP) and the average author purity (AAP). Given an ambiguous group, ACP evaluates the purity of the empirical clusters with respect to the theoretical clusters for this ambiguous group. Thus, if the empirical clusters are pure (i.e., they contain only references associated with the same author), the

[6]It is available at *http://www.lbd.dcc.ufmg.br/lbd/collections/disambiguation*
[7]It is available at *http://www.kisti.re.kr*

corresponding ACP value will be 1. ACP is defined in Equation 5:

$$\text{ACP} = \frac{1}{N} \sum_{i=1}^{e} \sum_{j=1}^{t} \frac{n_{ij}^2}{n_i} \tag{5}$$

where N is the total number of references in the ambiguous group, t is the number of theoretical clusters in the ambiguous group, e is the number of empirical clusters for this ambiguous group, n_i is the total number of references in the empirical cluster i, and n_{ij} is the total number of references in the empirical cluster i which are also in the theoretical cluster j.

For a given ambiguous group, AAP evaluates the fragmentation of the empirical clusters with respect to the theoretical clusters. If the empirical clusters are not fragmented, the corresponding AAP value will be 1. AAP is defined in Equation 6:

$$\text{AAP} = \frac{1}{N} \sum_{j=1}^{t} \sum_{i=1}^{e} \frac{n_{ij}^2}{n_j} \tag{6}$$

where n_j is the total number of references in the theoretical cluster j.

The K metric consists of the geometric mean between ACP and AAP values. It evaluates the purity and cohesion of the empirical clusters extracted by each method. The K metric is given in Equation 7:

$$K = \sqrt{\text{ACP} \times \text{AAP}} \tag{7}$$

Pairwise F1

Pairwise F1 (pF1) is the F1 metric [Rijsbergen 1979] calculated using pairwise precision and pairwise recall. Pairwise precision (pP) is calculated as $pP = \frac{a}{a+c}$, where a is the number of pairs of references in an empirical cluster that are (correctly) associated with the same author, and c is the number of pairs of references in an empirical cluster not corresponding to the same author. Pairwise recall (pR) is calculated as $pR = \frac{a}{a+b}$, where b is the number of pairs of references associated with the same author that are not in the same empirical cluster. The F1-metric is defined in Equation 8:

$$p\text{F1} = 2 \times \frac{pP \times pR}{pP + pR} \tag{8}$$

5.4 Experimental setup

Experiments were conducted with each ambiguous group, considering author name, list of coauthor names, work title and publication venue title as attributes. The same preprocessing was applied to all baselines, by removing punctuation and stopwords of work and publication venue titles, and stemming work titles using Porter's algorithm [Porter 1980]. Additionally, all coauthor names were standardized by considering only the initial letter of the first name along with the full last name.

Each method was executed ten times. In each execution, the metrics for each ambiguous group were calculated and averaged for all groups. The presented result is the average of the class averages for all runs, with the corresponding standard deviations.

For SVM and NB, each ambiguous group was randomly split into training (50%) and testing (50%) sets. Our strategy with user relevance feedback were performed only with the testing sets. This split ensures that the supervised methods have enough data for training as well as a fair comparison with our strategy. All results shown below also correspond to the performance in the testing sets. The results are compared using statistical significance tests (t-test) with 99% confidence interval.

Table II. SANDReF performance in DBLP under K metric labeling p% of doubts.

Group	# of references	# of doubts	p=0%	p=20%	Gain	p=40%	Gain	p=60%	Gain	p=80%	Gain	p=100%	Gain
A. Gupta	288	14.3 (9.6)	0.71 (0.03)	0.73 (0.02)	2.6%	0.75 (0.03)	4.3%	0.76 (0.03)	6.8%	0.78 (0.02)	8.7%	0.78 (0.02)	9.1%
A. Kumar	121	8.8 (6.1)	0.63 (0.05)	0.65 (0.05)	3.6%	0.66 (0.05)	5.4%	0.68 (0.05)	8.6%	0.69 (0.05)	9.9%	0.69 (0.05)	10.0%
C. Chen	399	55.5 (23.3)	0.55 (0.01)	0.58 (0.01)	5.8%	0.60 (0.03)	9.3%	0.63 (0.03)	13.3%	0.64 (0.03)	15.8%	0.65 (0.04)	18.6%
D. Johnson	184	6.0 (2.8)	0.66 (0.04)	0.67 (0.04)	1.8%	0.67 (0.04)	2.4%	0.68 (0.04)	3.5%	0.68 (0.04)	4.0%	0.68 (0.04)	4.1%
J. Martin	56	0.9 (0.8)	0.82 (0.05)	0.83 (0.05)	0.7%	0.83 (0.05)	0.8%	0.83 (0.05)	0.8%	0.83 (0.05)	0.8%	0.83 (0.05)	0.8%
J. Robinson	85	3.7 (1.6)	0.72 (0.04)	0.72 (0.04)	1.1%	0.73 (0.04)	1.8%	0.73 (0.04)	2.1%	0.74 (0.04)	2.7%	0.74 (0.04)	2.7%
J. Smith	460	28.7 (14.0)	0.70 (0.03)	0.72 (0.03)	2.7%	0.73 (0.03)	3.9%	0.74 (0.03)	5.8%	0.76 (0.03)	7.7%	0.76 (0.03)	8.6%
K. Tanaka	140	9.0 (5.4)	0.73 (0.04)	0.78 (0.05)	5.9%	0.79 (0.05)	7.3%	0.79 (0.05)	8.3%	0.80 (0.05)	9.5%	0.81 (0.05)	9.9%
M. Brown	76	2.7 (1.1)	0.77 (0.04)	0.79 (0.04)	2.0%	0.80 (0.05)	3.3%	0.80 (0.05)	4.1%	0.80 (0.04)	4.1%	0.80 (0.04)	4.1%
M. Jones	130	5.1 (2.6)	0.74 (0.06)	0.75 (0.06)	1.4%	0.76 (0.06)	2.3%	0.76 (0.05)	3.5%	0.77 (0.05)	3.8%	0.77 (0.05)	4.4%
M. Miller	202	3.6 (1.7)	0.86 (0.02)	0.86 (0.02)	0.5%	0.87 (0.02)	1.2%	0.87 (0.02)	1.5%	0.87 (0.02)	1.5%	0.87 (0.02)	1.5%
Average	195	13	0.72 (0.01)	0.73 (0.01)	2.4%	0.74 (0.01)	3.6%	0.75 (0.01)	5.0%	0.76 (0.01)	5.8%	0.76 (0.01)	6.3%

Table III. SANDReF performance in DBLP under pF1 metric labeling p% of doubts.

Group	# of references	# of doubts	p=0%	p=20%	Gain	p=40%	Gain	p=60%	Gain	p=80%	Gain	p=100%	Gain
A. Gupta	288	14.3 (9.6)	0.68 (0.04)	0.70 (0.04)	3.0%	0.72 (0.04)	5.3%	0.74 (0.05)	8.3%	0.75 (0.04)	10.6%	0.76 (0.04)	10.9%
A. Kumar	121	8.8 (6.1)	0.42 (0.09)	0.45 (0.09)	8.1%	0.47 (0.09)	13.4%	0.50 (0.08)	21.2%	0.51 (0.08)	23.1%	0.52 (0.08)	24.0%
C. Chen	399	55.5 (23.3)	0.30 (0.03)	0.35 (0.05)	19.5%	0.38 (0.07)	30.0%	0.42 (0.07)	42.3%	0.45 (0.08)	51.7%	0.48 (0.09)	61.7%
D. Johnson	184	6.0 (2.8)	0.62 (0.05)	0.63 (0.06)	1.7%	0.64 (0.06)	2.2%	0.64 (0.06)	3.0%	0.64 (0.06)	3.2%	0.64 (0.06)	3.3%
J. Martin	56	0.9 (0.8)	0.68 (0.10)	0.69 (0.09)	1.8%	0.69 (0.09)	2.1%	0.69 (0.09)	2.1%	0.69 (0.09)	2.1%	0.69 (0.09)	2.1%
J. Robinson	85	3.7 (1.6)	0.65 (0.07)	0.66 (0.07)	1.6%	0.66 (0.07)	2.7%	0.67 (0.07)	2.9%	0.67 (0.07)	3.8%	0.67 (0.07)	3.8%
J. Smith	460	28.7 (14.0)	0.70 (0.04)	0.72 (0.04)	3.2%	0.73 (0.04)	4.4%	0.74 (0.04)	6.6%	0.76 (0.04)	8.9%	0.77 (0.04)	10.3%
K. Tanaka	140	9.0 (5.4)	0.62 (0.05)	0.69 (0.09)	12.4%	0.71 (0.08)	15.3%	0.72 (0.08)	17.0%	0.73 (0.08)	19.0%	0.74 (0.08)	19.7%
M. Brown	76	2.7 (1.1)	0.71 (0.08)	0.73 (0.08)	2.9%	0.75 (0.09)	5.0%	0.76 (0.09)	6.2%	0.76 (0.08)	6.2%	0.76 (0.08)	6.2%
M. Jones	130	5.1 (2.6)	0.69 (0.09)	0.71 (0.09)	1.9%	0.72 (0.08)	3.6%	0.73 (0.07)	5.2%	0.73 (0.08)	5.8%	0.74 (0.08)	6.7%
M. Miller	202	3.6 (1.7)	0.86 (0.03)	0.86 (0.03)	0.6%	0.87 (0.03)	1.1%	0.87 (0.03)	1.3%	0.87 (0.03)	1.3%	0.87 (0.03)	1.3%
Average	195	13	0.63 (0.02)	0.66 (0.03)	4.1%	0.67 (0.03)	6.1%	0.68 (0.03)	8.1%	0.69 (0.02)	9.5%	0.69 (0.03)	10.3%

In our work, we used the SVM implementation provided by the LibSVM package [Chang and Lin 2001], with RBF (Radial Basis Function) as the kernel function, where the best γ and cost values were obtained from the *Grid* program, available with the LibSVM package, for each training data. LibSVM uses the "one-against-one" and voting for multi-class classification [Chang and Lin 2001]. In our experiments, we used TF-IDF (term frequency-inverse document frequency) as feature weight scheme. We used a non-parametric implementation for Naïve Bayes [Domingos and Pazzani 1997].

For LaSVM-DBSCAN, we used the LaSVM package [Bordes et al. 2005][8] with RBF as the kernel function using the default values for cost and γ, respectively, and the DBSCAN version available from Weka[9] with the values of 1.5 and 2 for ϵ and the minimum number of points, respectively. We should stress that parameter tuning with cross-validation in the training was also performed for LaSVM but results produced by the default parameters turned out to be the best.

Hereafter we will refer to our strategy to author name disambiguation with user relevance feedback as SANDReF (standing for Self-training Author Name Disambiguation with Relevance Feedback). For it, we used the same parameters (e.g., ϕ_{max}) as defined in the original work that proposed SAND [Ferreira et al. 2010].

5.5 Results

The effectiveness of SANDReF. To evaluate the effectiveness of SANDReF and also the labeling effort needed to improve the disambiguation process, we take a percentage x, varying from 0% to 100%, of the first references with doubtful predictions from $\mathcal{U}'$ ordered by $\Delta(x)$ and execute the method. p=0% corresponds to the SANDReF performance without user relevance feedback, i.e., the original implementation of SAND. Tables II and III show the SANDReF performance in DBLP under K and pF1 metrics, respectively. In both tables, the column "# of doubts" corresponds to the average number of doubtful predictions obtained by the first round of the supervised step (line 2 of Algorithm 1). The column titles p=x% indicate the percentage of references manually labeled from the doubtful predictions. Each column named *Gain*, shows the percentage gains of SANDReF of p=x% over p=0%.

[8] Available at *http://leon.bottou.org/research/lasvm*
[9] *http://www.cs.waikato.ac.nz/ml/weka/*

Table IV. SANDReF performance in KISTI under K metric labeling p% of doubts.

Group	# of references	# of doubts	p=0%	p=20%	Gain	p=40%	Gain	p=60%	Gain	p=80%	Gain	p=100%	Gain
A. Choudhary	80	0.1 (0.3)	0.89 (0.03)	0.90 (0.03)	0.1%	0.90 (0.03)	0.1%	0.90 (0.03)	0.1%	0.90 (0.03)	0.1%	0.90 (0.03)	0.1%
A. Gupta	83	5.1 (2.8)	0.76 (0.04)	0.80 (0.05)	5.3%	0.82 (0.05)	7.0%	0.82 (0.06)	8.2%	0.83 (0.05)	8.6%	0.83 (0.05)	8.8%
D. Eppstein	98	12.5 (4.8)	0.44 (0.16)	0.49 (0.18)	12.1%	0.52 (0.19)	17.4%	0.53 (0.20)	20.9%	0.55 (0.20)	24.1%	0.56 (0.21)	26.3%
D. Lee	93	6.3 (3.7)	0.69 (0.24)	0.70 (0.25)	1.7%	0.71 (0.25)	3.1%	0.71 (0.25)	3.5%	0.71 (0.25)	3.6%	0.71 (0.25)	3.8%
H. Chen	75	5.8 (2.2)	0.72 (0.02)	0.73 (0.02)	2.2%	0.74 (0.03)	3.8%	0.74 (0.03)	3.8%	0.75 (0.03)	4.4%	0.75 (0.03)	5.1%
H. Wang	134	9.0 (5.9)	0.72 (0.26)	0.73 (0.26)	0.5%	0.74 (0.26)	2.6%	0.75 (0.27)	3.3%	0.76 (0.27)	4.4%	0.76 (0.27)	4.9%
J. Chen	126	10.1 (4.5)	0.66 (0.23)	0.67 (0.24)	1.1%	0.67 (0.24)	2.3%	0.69 (0.24)	4.3%	0.70 (0.25)	5.7%	0.70 (0.25)	5.9%
J. Dongarra	73	1.5 (1.4)	0.78 (0.07)	0.80 (0.08)	1.8%	0.80 (0.08)	1.8%	0.80 (0.08)	1.8%	0.80 (0.08)	1.8%	0.80 (0.08)	1.9%
J. Halpern	118	10.3 (8.0)	0.51 (0.19)	0.53 (0.20)	4.7%	0.56 (0.20)	8.9%	0.57 (0.21)	11.9%	0.58 (0.21)	13.3%	0.58 (0.21)	14.0%
J. Kim	74	3.6 (2.2)	0.82 (0.03)	0.82 (0.03)	0.8%	0.82 (0.03)	0.9%	0.83 (0.03)	1.7%	0.83 (0.03)	1.7%	0.83 (0.03)	2.2%
J. Lee	101	8.2 (4.4)	0.73 (0.26)	0.74 (0.26)	2.0%	0.75 (0.26)	3.0%	0.75 (0.27)	3.6%	0.76 (0.27)	4.3%	0.76 (0.27)	4.6%
J. Li	87	5.7 (1.8)	0.76 (0.27)	0.77 (0.27)	1.2%	0.78 (0.28)	2.4%	0.79 (0.28)	3.3%	0.79 (0.28)	3.7%	0.79 (0.28)	3.9%
J. Liu	85	7.2 (4.0)	0.69 (0.25)	0.70 (0.25)	1.4%	0.72 (0.25)	4.1%	0.72 (0.25)	4.4%	0.72 (0.25)	4.9%	0.72 (0.25)	4.9%
J. Mitchell	85	3.5 (1.8)	0.61 (0.22)	0.61 (0.22)	1.2%	0.63 (0.22)	3.4%	0.64 (0.23)	5.2%	0.64 (0.23)	6.1%	0.64 (0.23)	6.1%
J. Smith	80	4.4 (1.4)	0.79 (0.04)	0.80 (0.04)	1.2%	0.81 (0.05)	2.7%	0.82 (0.05)	3.4%	0.82 (0.05)	4.0%	0.82 (0.05)	4.1%
J. Wang	162	11.6 (4.4)	0.68 (0.24)	0.69 (0.24)	1.8%	0.70 (0.25)	3.0%	0.71 (0.25)	4.4%	0.72 (0.25)	5.8%	0.72 (0.25)	5.9%
J. Wu	95	2.9 (1.9)	0.69 (0.25)	0.70 (0.25)	2.1%	0.71 (0.25)	2.5%	0.71 (0.25)	3.1%	0.71 (0.25)	3.6%	0.71 (0.25)	3.6%
J. Zhang	101	7.7 (1.9)	0.73 (0.26)	0.74 (0.26)	1.2%	0.76 (0.27)	3.1%	0.77 (0.27)	4.6%	0.77 (0.27)	5.3%	0.78 (0.27)	5.8%
L. Zhang	93	4.4 (1.8)	0.72 (0.25)	0.72 (0.26)	0.9%	0.73 (0.26)	1.3%	0.73 (0.26)	1.9%	0.73 (0.26)	2.5%	0.74 (0.26)	2.7%
M. Chen	94	6.5 (2.3)	0.62 (0.22)	0.63 (0.23)	2.4%	0.64 (0.23)	3.0%	0.65 (0.23)	4.7%	0.65 (0.24)	5.3%	0.65 (0.24)	5.3%
M. Pedram	79	5.0 (3.8)	0.71 (0.11)	0.72 (0.10)	2.7%	0.74 (0.10)	5.4%	0.77 (0.08)	9.0%	0.78 (0.06)	10.2%	0.78 (0.06)	10.6%
M. Vardi	88	5.8 (3.8)	0.58 (0.21)	0.59 (0.21)	2.5%	0.60 (0.22)	4.5%	0.61 (0.22)	5.2%	0.61 (0.22)	6.5%	0.61 (0.22)	6.7%
N. Jennings	75	4.0 (3.7)	0.67 (0.11)	0.70 (0.10)	4.9%	0.71 (0.10)	6.0%	0.71 (0.10)	6.0%	0.71 (0.09)	6.6%	0.71 (0.09)	6.7%
N. Jha	89	0.0 (0.0)	0.84 (0.29)	0.84 (0.29)	0.0%	0.84 (0.29)	0.0%	0.84 (0.29)	0.0%	0.84 (0.29)	0.0%	0.84 (0.29)	0.0%
N. Lynch	82	6.5 (4.4)	0.58 (0.11)	0.61 (0.11)	5.1%	0.62 (0.11)	6.6%	0.63 (0.11)	8.7%	0.64 (0.09)	10.4%	0.65 (0.09)	11.7%
P. Yu	83	0.5 (0.5)	0.82 (0.29)	0.82 (0.29)	0.6%	0.82 (0.29)	0.6%	0.82 (0.29)	0.6%	0.82 (0.29)	0.6%	0.82 (0.29)	0.6%
Q. Yang	79	4.6 (2.1)	0.67 (0.04)	0.69 (0.04)	3.0%	0.70 (0.03)	4.7%	0.71 (0.03)	6.2%	0.72 (0.03)	7.6%	0.72 (0.03)	7.8%
S. Jajodia	109	0.7 (0.9)	0.77 (0.27)	0.78 (0.28)	0.3%	0.78 (0.28)	0.3%	0.78 (0.28)	0.3%	0.78 (0.28)	0.3%	0.78 (0.28)	0.3%
S. Kim	78	3.8 (2.0)	0.87 (0.03)	0.88 (0.03)	0.6%	0.89 (0.02)	1.4%	0.89 (0.03)	1.7%	0.89 (0.03)	2.1%	0.89 (0.02)	2.4%
S. Lee	88	5.8 (3.6)	0.71 (0.25)	0.72 (0.25)	1.0%	0.72 (0.25)	1.4%	0.72 (0.26)	1.8%	0.74 (0.26)	3.7%	0.74 (0.26)	4.3%
T. Henzinger	87	0.1 (0.3)	0.81 (0.29)	0.81 (0.29)	0.0%	0.81 (0.29)	0.0%	0.81 (0.29)	0.0%	0.81 (0.29)	0.0%	0.81 (0.29)	0.0%
W. Wang	89	1.8 (2.2)	0.70 (0.25)	0.71 (0.25)	1.3%	0.71 (0.25)	1.6%	0.71 (0.25)	1.8%	0.72 (0.25)	2.0%	0.72 (0.25)	2.0%
X. Li	152	10.5 (4.8)	0.66 (0.23)	0.67 (0.24)	1.2%	0.68 (0.24)	3.2%	0.69 (0.24)	4.2%	0.69 (0.24)	4.8%	0.69 (0.24)	5.0%
X. Zhou	74	1.3 (1.3)	0.90 (0.04)	0.91 (0.04)	1.0%	0.91 (0.04)	1.6%	0.92 (0.04)	1.8%	0.92 (0.04)	1.9%	0.92 (0.04)	1.9%
Y. Chen	151	10.3 (4.1)	0.66 (0.23)	0.67 (0.24)	1.2%	0.68 (0.24)	2.4%	0.68 (0.24)	3.6%	0.70 (0.24)	5.2%	0.70 (0.25)	5.7%
Y. Liu	156	9.6 (5.1)	0.72 (0.26)	0.74 (0.26)	1.8%	0.74 (0.26)	2.8%	0.76 (0.27)	4.4%	0.76 (0.27)	4.4%	0.76 (0.27)	4.8%
Y. Wang	128	11.7 (5.0)	0.67 (0.24)	0.69 (0.24)	2.6%	0.69 (0.24)	3.9%	0.70 (0.25)	5.0%	0.71 (0.25)	6.1%	0.71 (0.25)	6.8%
Y. Yang	82	5.5 (2.0)	0.77 (0.06)	0.78 (0.05)	2.0%	0.79 (0.05)	2.6%	0.79 (0.04)	3.3%	0.80 (0.04)	4.1%	0.80 (0.05)	4.3%
Y. Zhang	104	6.5 (2.2)	0.69 (0.24)	0.70 (0.25)	1.2%	0.71 (0.25)	2.7%	0.71 (0.25)	4.0%	0.72 (0.25)	4.7%	0.72 (0.26)	5.0%
Z. Zhang	103	5.1 (3.1)	0.74 (0.26)	0.75 (0.27)	1.4%	0.75 (0.27)	1.7%	0.75 (0.27)	1.6%	0.75 (0.27)	1.5%	0.75 (0.27)	2.2%
Average	98	6	0.76 (0.01)	0.77 (0.01)	1.8%	0.79 (0.01)	2.9%	0.79 (0.01)	3.8%	0.80 (0.01)	4.4%	0.80 (0.01)	4.7%

In Table II and III, we can notice that labeling all references (p=100%) with doubtful predictions produces gains of up to 6.3% and 10.3% under K and pF1 metrics, respectively, on average. This corresponds to labeling around 6%[10]of the total number of references[11]. We can also see that labeling around 80% of the references with doubtful predictions (5% of the total number of references) the gains are very similar, around 5.8% and 9.5% under K and pF1 metrics. Notice also that even if we label only 20% of the doubts (about 1.2% of the total number of references), we still can get up to 4.1% of improvement in pF1. Finally, by looking at specific ambiguous groups, we can see some gains are quite impressive: up to 61.7% for the "C. Chen" group and 19.7% for the "J. Smith" group in terms of pF1. These are exactly the largest and most difficult groups to disambiguate in this collection.

Tables IV and V show the SANDReF performance under K and pF1 metrics in KISTI collection. We also notice that labeling about 80% of the doubtful predictions (around 4.5% of the total number of references in this collection), the gains are also close to those when we label all doubtful predictions: around 4.5% and 8.5% under K and pF1 metrics, respectively. We also see a similar trend for higher gains in some large and more ambiguous groups (e.g., 44.9% of gains in the "D. Eppstein" and 26.1% for the "J. Chen" ambiguous groups).

Comparing SANDReF with Supervised Methods. Here we compare SANDReF with four supervised author assignment and grouping methods, namely, SVM, NB, LaSVM-DBSCAN and SSAND which have historically produced some of the best performances in the author name disambiguation task. Particularly, SSAND corresponds to the supervised step of SAND. As said before, the supervised

[10]We believe that this is very reasonable amount as we are talking about 6% of the ambiguous references, which is only a subset of all entries in a DL.

[11]Notice that labeling this percentage of instances does not guarantee the same percentage of improvement, as some of the doubts could be in fact correctly predicted in the end of the process of the original method without feedback.

Table V. SANDReF performance in KISTI under pF1 metric labeling p% of doubts.

Group	# of references	# of doubts	p=0%	p=20%	Gain	p=40%	Gain	p=60%	Gain	p=80%	Gain	p=100%	Gain
A. Choudhary	80	0.1 (0.3)	0.89 (0.03)	0.89 (0.03)	0.2%	0.89 (0.03)	0.2%	0.89 (0.03)	0.2%	0.89 (0.03)	0.2%	0.89 (0.03)	0.2%
A. Gupta	83	5.1 (2.8)	0.73 (0.06)	0.79 (0.06)	7.2%	0.80 (0.06)	9.2%	0.81 (0.07)	10.3%	0.81 (0.06)	11.0%	0.81 (0.06)	11.2%
D. Eppstein	98	12.5 (4.8)	0.34 (0.14)	0.41 (0.16)	20.5%	0.44 (0.18)	29.5%	0.46 (0.18)	35.6%	0.48 (0.19)	41.0%	0.49 (0.19)	44.8%
D. Lee	93	6.3 (3.7)	0.56 (0.20)	0.58 (0.20)	3.3%	0.59 (0.21)	5.3%	0.59 (0.21)	6.2%	0.59 (0.21)	6.5%	0.60 (0.21)	6.9%
H. Chen	75	5.8 (2.2)	0.40 (0.07)	0.46 (0.09)	15.2%	0.49 (0.08)	23.0%	0.51 (0.10)	25.8%	0.51 (0.11)	27.8%	0.52 (0.10)	30.1%
H. Wang	134	9.0 (5.9)	0.68 (0.24)	0.68 (0.24)	0.5%	0.71 (0.26)	3.5%	0.71 (0.26)	4.3%	0.72 (0.26)	6.1%	0.73 (0.27)	6.9%
J. Chen	126	10.1 (4.5)	0.37 (0.15)	0.38 (0.15)	2.1%	0.41 (0.15)	9.1%	0.45 (0.18)	20.9%	0.47 (0.18)	25.5%	0.47 (0.18)	26.1%
J. Dongarra	73	1.5 (1.4)	0.75 (0.09)	0.77 (0.10)	2.3%	0.77 (0.10)	2.3%	0.77 (0.10)	2.3%	0.77 (0.10)	2.3%	0.77 (0.09)	2.4%
J. Halpern	118	10.3 (8.0)	0.46 (0.18)	0.49 (0.19)	6.8%	0.52 (0.20)	13.3%	0.54 (0.20)	17.5%	0.55 (0.20)	19.6%	0.55 (0.20)	20.3%
J. Kim	74	3.6 (2.2)	0.53 (0.05)	0.55 (0.05)	3.6%	0.55 (0.05)	4.3%	0.56 (0.04)	5.6%	0.55 (0.07)	4.7%	0.56 (0.07)	5.7%
J. Lee	101	8.2 (4.4)	0.56 (0.21)	0.58 (0.22)	4.8%	0.60 (0.22)	7.9%	0.61 (0.22)	10.0%	0.63 (0.23)	13.0%	0.63 (0.23)	14.0%
J. Li	87	5.7 (1.8)	0.68 (0.25)	0.69 (0.26)	2.4%	0.71 (0.26)	4.6%	0.71 (0.26)	5.8%	0.73 (0.27)	7.6%	0.73 (0.27)	8.1%
J. Liu	85	7.2 (4.0)	0.52 (0.20)	0.53 (0.20)	1.0%	0.56 (0.21)	7.3%	0.57 (0.21)	8.9%	0.58 (0.21)	11.0%	0.58 (0.21)	11.0%
J. Mitchell	85	3.5 (1.8)	0.53 (0.20)	0.54 (0.20)	2.1%	0.56 (0.20)	5.8%	0.57 (0.21)	8.8%	0.58 (0.21)	10.5%	0.58 (0.21)	10.5%
J. Smith	80	4.4 (1.4)	0.75 (0.09)	0.77 (0.10)	2.0%	0.78 (0.10)	3.9%	0.79 (0.10)	5.1%	0.80 (0.09)	6.3%	0.81 (0.09)	6.8%
J. Wang	162	11.6 (4.4)	0.60 (0.22)	0.62 (0.23)	2.6%	0.63 (0.23)	4.8%	0.65 (0.23)	7.3%	0.67 (0.24)	10.5%	0.67 (0.24)	10.4%
J. Wu	95	2.9 (1.9)	0.63 (0.24)	0.65 (0.24)	3.8%	0.65 (0.24)	4.7%	0.66 (0.24)	5.6%	0.67 (0.24)	6.6%	0.67 (0.24)	6.6%
J. Zhang	101	7.7 (1.9)	0.62 (0.23)	0.64 (0.23)	2.5%	0.66 (0.24)	6.5%	0.69 (0.25)	10.6%	0.70 (0.25)	12.5%	0.71 (0.25)	13.8%
L. Zhang	93	4.4 (1.8)	0.61 (0.23)	0.61 (0.23)	-0.1%	0.63 (0.23)	2.6%	0.64 (0.23)	3.8%	0.64 (0.24)	4.6%	0.64 (0.24)	4.8%
M. Chen	94	6.5 (2.3)	0.48 (0.21)	0.50 (0.22)	4.7%	0.50 (0.22)	5.2%	0.52 (0.23)	8.4%	0.52 (0.23)	9.5%	0.52 (0.23)	8.7%
M. Pedram	79	5.0 (3.8)	0.66 (0.15)	0.68 (0.13)	3.8%	0.70 (0.13)	7.5%	0.74 (0.09)	12.5%	0.75 (0.08)	14.1%	0.75 (0.07)	14.7%
M. Vardi	88	5.8 (3.8)	0.51 (0.19)	0.53 (0.20)	3.8%	0.55 (0.21)	6.8%	0.55 (0.21)	7.8%	0.56 (0.21)	9.8%	0.57 (0.21)	10.0%
N. Jennings	75	4.0 (3.7)	0.61 (0.15)	0.65 (0.13)	7.0%	0.66 (0.12)	8.7%	0.66 (0.13)	8.6%	0.66 (0.12)	9.5%	0.67 (0.12)	9.6%
N. Jha	89	0.0 (0.0)	0.83 (0.29)	0.83 (0.29)	0.0%	0.83 (0.29)	0.0%	0.83 (0.29)	0.0%	0.83 (0.29)	0.0%	0.83 (0.29)	0.0%
N. Lynch	82	6.5 (4.4)	0.50 (0.14)	0.54 (0.14)	7.9%	0.55 (0.14)	10.1%	0.56 (0.14)	13.3%	0.58 (0.12)	15.9%	0.58 (0.11)	17.8%
P. Yu	83	0.5 (0.5)	0.81 (0.29)	0.81 (0.29)	0.9%	0.81 (0.29)	0.9%	0.81 (0.29)	0.9%	0.81 (0.29)	0.9%	0.81 (0.29)	0.9%
Q. Yang	79	4.6 (2.1)	0.57 (0.07)	0.60 (0.07)	5.4%	0.61 (0.07)	7.0%	0.63 (0.07)	9.5%	0.64 (0.06)	12.6%	0.65 (0.06)	12.9%
S. Jajodia	109	0.7 (0.9)	0.78 (0.28)	0.78 (0.28)	0.4%	0.78 (0.28)	0.4%	0.78 (0.28)	0.4%	0.78 (0.28)	0.4%	0.78 (0.28)	0.4%
S. Kim	78	3.8 (2.0)	0.81 (0.07)	0.81 (0.07)	0.4%	0.84 (0.06)	3.2%	0.84 (0.06)	3.0%	0.84 (0.06)	3.4%	0.84 (0.05)	3.8%
S. Lee	88	5.8 (3.6)	0.52 (0.20)	0.53 (0.21)	2.7%	0.55 (0.21)	5.6%	0.56 (0.21)	7.1%	0.58 (0.23)	12.4%	0.59 (0.23)	13.7%
T. Henzinger	87	0.1 (0.3)	0.81 (0.28)	0.81 (0.28)	0.0%	0.81 (0.28)	0.0%	0.81 (0.28)	0.0%	0.81 (0.28)	0.0%	0.81 (0.28)	0.0%
W. Wang	89	1.8 (2.2)	0.56 (0.20)	0.57 (0.21)	1.9%	0.58 (0.21)	2.3%	0.58 (0.21)	2.6%	0.58 (0.21)	3.4%	0.58 (0.21)	3.4%
X. Li	152	10.5 (4.8)	0.53 (0.19)	0.54 (0.19)	1.9%	0.56 (0.20)	5.9%	0.58 (0.21)	8.5%	0.58 (0.21)	9.4%	0.58 (0.21)	9.5%
X. Zhou	74	1.3 (1.3)	0.86 (0.06)	0.88 (0.06)	1.7%	0.88 (0.06)	2.4%	0.88 (0.06)	2.6%	0.89 (0.06)	2.8%	0.89 (0.06)	2.8%
Y. Chen	151	10.3 (4.1)	0.50 (0.19)	0.51 (0.19)	1.9%	0.52 (0.19)	4.4%	0.53 (0.19)	7.3%	0.55 (0.20)	10.9%	0.56 (0.20)	12.3%
Y. Liu	156	9.6 (5.1)	0.66 (0.24)	0.69 (0.25)	4.0%	0.70 (0.25)	5.9%	0.72 (0.26)	8.6%	0.72 (0.26)	9.2%	0.73 (0.26)	10.0%
Y. Wang	128	11.7 (5.0)	0.49 (0.18)	0.51 (0.18)	5.3%	0.54 (0.20)	11.2%	0.56 (0.21)	14.1%	0.58 (0.22)	18.0%	0.58 (0.22)	19.6%
Y. Yang	82	5.5 (2.0)	0.61 (0.12)	0.63 (0.12)	4.0%	0.65 (0.12)	6.1%	0.66 (0.11)	7.6%	0.67 (0.12)	9.9%	0.67 (0.12)	10.2%
Y. Zhang	104	6.5 (2.2)	0.52 (0.19)	0.53 (0.19)	2.3%	0.54 (0.20)	5.7%	0.56 (0.20)	7.8%	0.57 (0.21)	9.7%	0.57 (0.21)	10.1%
Z. Zhang	103	5.1 (3.1)	0.64 (0.24)	0.65 (0.24)	0.8%	0.65 (0.24)	0.7%	0.65 (0.24)	0.5%	0.65 (0.24)	1.2%	0.66 (0.24)	2.2%
Average pF1	98	6	0.65 (0.02)	0.67 (0.02)	3.1%	0.69 (0.02)	5.5%	0.70 (0.02)	7.0%	0.71 (0.02)	8.5%	0.71 (0.02)	8.9%

author assignment methods use 50% of the references as training data and other 50% as test set and the performance of all methods corresponds to the performance on test set. The SANDReF performance corresponds to the performance after labeling all references with doubtful predictions (p=100%).

Table VI. Comparison with supervised author grouping and assignment methods in DBLP and KISTI. The best results are highlighted in bold.

Collection	Method	K	pF1
DBLP	SANDReF	0.76 (0.01)	0.69 (0.03)
	SSAND	**0.88** (0.01)	**0.87** (0.01)
	SVM	0.80 (0.01)	0.72 (0.01)
	NB	0.74 (0.01)	0.65 (0.01)
	LaSVM-DBSCAN	0.61 (0.02)	0.44 (0.02)
KISTI	SANDReF	0.80 (0.01)	0.71 (0.02)
	SSAND	**0.90** (0.00)	**0.86** (0.01)
	LaSVM-DBSCAN	0.76 (0.01)	0.60 (0.01)
	SVM	0.76 (0.00)	0.58 (0.01)
	NB	0.75 (0.00)	0.60 (0.01)

We can notice that the supervised step of SAND using the same training data of the other supervised methods (SSAND) produces the best results. However, as mentioned, SSAND requires the initial labeling of half of the collection. In DBLP, SVM outperforms SANDReF and NB, but SANDReF is only 4.6% and 3.7% worse than SVM under the K and pF1 metrics, respectively, without using any initial training data. SANDReF also outperforms NB and LaSVM-DBSCAN, in this collection, this last one by a large margin.

In KISTI, SANDReF is the second best performer among all methods, being around 5% and 21% better than LaSVM-DBSCAN and SVM, under the K and pF1 metrics, and with gains of more than 7% and 18% over NB, under the same metrics.

Table VII. SANDReF performance in DBLP under K metric labeling p% of doubts and performing the supervised step on references with doubtful predictions.

Group	p=0%	p=20%	Gain	p=40%	Gain	p=60%	Gain	p=80%	Gain	p=100%	Gain
A. Gupta	0.71 (0.04)	0.71 (0.04)	0.1%	0.72 (0.04)	1.1%	0.73 (0.04)	2.1%	0.73 (0.04)	2.8%	0.73 (0.04)	2.9%
A. Kumar	0.64 (0.05)	0.64 (0.04)	0.7%	0.65 (0.04)	1.5%	0.66 (0.04)	2.9%	0.66 (0.04)	3.3%	0.66 (0.04)	3.4%
C. Chen	0.55 (0.02)	0.57 (0.02)	2.3%	0.59 (0.03)	5.9%	0.60 (0.03)	8.6%	0.62 (0.04)	11.4%	0.63 (0.05)	13.5%
D. Johnson	0.67 (0.04)	0.67 (0.04)	0.9%	0.68 (0.04)	2.3%	0.68 (0.04)	2.5%	0.68 (0.04)	2.9%	0.68 (0.04)	3.0%
J. Martin	0.82 (0.03)	0.83 (0.03)	1.1%	0.83 (0.03)	1.3%	0.83 (0.03)	1.3%	0.83 (0.03)	1.3%	0.83 (0.03)	1.3%
J. Robinson	0.73 (0.03)	0.73 (0.03)	-0.1%	0.74 (0.03)	1.0%	0.75 (0.04)	2.1%	0.75 (0.04)	2.5%	0.75 (0.04)	2.5%
J. Smith	0.70 (0.03)	0.70 (0.03)	0.6%	0.71 (0.03)	1.4%	0.72 (0.03)	2.7%	0.72 (0.03)	3.6%	0.73 (0.03)	4.4%
K. Tanaka	0.73 (0.04)	0.73 (0.04)	0.6%	0.74 (0.04)	1.6%	0.75 (0.04)	2.8%	0.75 (0.04)	3.5%	0.75 (0.04)	3.7%
M. Brown	0.76 (0.03)	0.76 (0.03)	0.4%	0.77 (0.03)	1.0%	0.77 (0.02)	1.2%	0.77 (0.02)	1.4%	0.77 (0.02)	1.4%
M. Jones	0.74 (0.07)	0.75 (0.07)	1.6%	0.76 (0.07)	2.1%	0.76 (0.07)	2.6%	0.76 (0.07)	2.7%	0.76 (0.07)	2.8%
M. Miller	0.74 (0.09)	0.74 (0.09)	-0.1%	0.75 (0.09)	0.3%	0.75 (0.09)	0.5%	0.75 (0.09)	0.7%	0.75 (0.09)	0.9%

In sum, our proposed method is very competitive when compared to the supervised author grouping and even with most supervised author assignment ones, outperforming some of them by large margins like in KISTI, loosing only to SSAND, which has the cost of having to label half of the collection to obtain such gains, against 6% of our method. Therefore if the cost of labeling is very high for a given DL, our method would be a good choice.

Checking whether the labeling helps the prediction. One issue that remains to be investigated is how much of the observed gains are really coming from information being obtained from these doubtful cases and used in the models learned for the supervised step of our method and how much is from the actual labeling of these instances.

To answer this question, we performed an additional experiment, in DBLP predicting only the references in $\mathcal{U} - \mathcal{U}'$ (i.e., references with doubtful predictions that were not labeled and inserted into the training data $\mathcal{D}'$) instead of predicting all test set $\mathcal{T}'$ in the second round of the supervised step. For instance, with p=60%, we label 60% of the doubtful cases and use $\mathcal{D}'$ to assign authors to the remaining 40%. The case of p=100% corresponds to the situation in which we label all doubts and this would be the maximum gain that could be obtained without re-running the supervised step in the test set of the method, as we did in the previous experiments. Table VII shows the results of this experiment.

Comparing the results from Table VII with the ones from Table II, mainly the case of p=100%, we can see that all gains are much higher in Table II than in Table VII in all ambiguous groups meaning that the labeling of the doubtful cases really helped to correctly assign some references that otherwise would be incorrectly predicted without the information coming from these additional labeled instances.

Regarding efficiency issues of proposed method, we measured the time spent to infer the author of each reference in the test set. On average, the time to assign each reference to its author was around 0.2 second. We perform our evaluation in a Intel Xeon E5620 with 2.40GHz and 8 gigabytes of RAM. This means that we could disambiguate approximately half million records in one day.

6. CONCLUSIONS

In this article, we exploited user relevance feedback in the author name disambiguation task. We propose to select a very small portion of the references for which our method was not sure about the correct authorship and ask the administrators for labeling them. These labeled references are then incorporated into the training data automatically selected by our self-training author name method and used by the supervised step of our method for disambiguation. In our experimental evaluation, we observed that with a very small labeling effort (usually around 5% of all DL records), the disambiguation effectiveness improved by almost 10% on average in a collection under the pairwise

F1 metric, with gains of up to 61% in some of the most difficult ambiguous groups.

In future work, we intend to explore issues related to the Δ_{min} parameter in order to better investigate the tradeoffs between labeling and effectiveness gains. We also want to study the impact of using several rounds of relevance feedback and the corresponding tradeoffs (since labeling costs tend to increase with more rounds) as well as to better investigate why certain predictions made with certainty were wrongly assigned.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. Washington-DC, USA, pp. 207–216, 1993.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley Publishing Company, 2008.

Bhattacharya, I. and Getoor, L. A latent dirichlet model for unsupervised entity resolution. In *Proceedings of the Sixth SIAM International Conference on Data Mining*. Bethesda, MD, USA, 2006.

Bhattacharya, I. and Getoor, L. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data* 1 (1): Article 5, 2007.

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research* 6 (9): 1579–1619, 2005.

Carvalho, A. P., Ferreira, A. A., Laender, A. H. F., and Gonçalves, M. A. Incremental unsupervised name disambiguation in cleaned digital libraries. *Journal of Information and Data Management* 2 (3): 289–304, 2011.

Chang, C.-C. and Lin, C.-J. *LibSVM: A Library for Support Vector Machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Cota, R. G., Ferreira, A. A., Gonçalves, M. A., Laender, A. H. F., and Nascimento, C. An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. *Journal of the American Society for Information Science and Technology* 61 (9): 1853–1870, 2010.

Cota, R. G., Gonçalves, M. A., and Laender, A. H. F. A heuristic-based hierarchical clustering method for author name disambiguation in digital libraries. In *Proceedings of the XXII Brazilian Symposium on Databases*. João Pessoa, Paraiba, Brazil, pp. 20–34, 2007.

Culotta, A., Kanani, P., Hall, R., Wick, M., and McCallum, A. Author disambiguation using error-driven machine learning with a ranking loss function. In *Proceedings of the International Workshop on Information Integration on the Web*. Vancouver, Canada, 2007.

Domingos, P. and Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* vol. 29, pp. 103–137, 1997.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon, pp. 226–231, 1996.

Fan, X., Wang, J., Pu, X., Zhou, L., and Lv, B. On graph-based name disambiguation. *ACM Journal of Data and Information Quality* vol. 2, pp. 10:1–10:23, February, 2011.

Ferreira, A. A. *Contributions for Solving the Author Name Ambiguity Problem in Bibliographic Citations*. Ph.D. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2012.

Ferreira, A. A., Gonçalves, M. A., and Laender, A. H. F. A brief survey of automatic methods for author name disambiguation. *SIGMOD Record* 41 (2): 15–26, 2012a.

Ferreira, A. A., Silva, R., Gonçalves, M. A., Veloso, A., and Laender, A. H. F. Active associative sampling for author name disambiguation. In *Proceedings of the 2012 ACM/IEEE Joint Conference on Digital Libraries*. Washington DC, pp. 175–184, 2012b.

Ferreira, A. A., Veloso, A., Gonçalves, M. A., and Laender, A. H. F. Effective self-training author name disambiguation in scholarly digital libraries. In *Proceedings of the 2010 ACM/IEEE Joint Conference on Digital Libraries*. Gold Coast, Queensland, Australia, pp. 39–48, 2010.

Geisser, S. *Predictive inference: An introduction*. New York, 1993.

Han, H., Giles, C. L., Zha, H., Li, C., and Tsioutsiouliklis, K. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*. Tuscon, USA, pp. 296–305, 2004.

Han, H., Xu, W., Zha, H., and Giles, C. L. A hierarchical naive Bayes mixture model for name disambiguation in author citations. In *Proceedings of the 2005 ACM Symposium on Applied Computing*. Santa Fe, New Mexico, USA, pp. 1065–1069, 2005.

Han, H., Zha, H., and Giles, C. L. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries*. Denver, CO, USA, pp. 334–343, 2005.

Huang, J., Ertekin, S., and Giles, C. L. Efficient name disambiguation for large-scale databases. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Berlin, Germany, pp. 536–544, 2006.

Kanani, P., McCallum, A., and Pal, C. Improving author coreference by resource-bounded information gathering from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. Hyderabad, India, pp. 429–434, 2007.

Kang, I.-S., Kim, P., Lee, S., Jung, H., and You, B.-J. Construction of a large-scale test set for author disambiguation. *Information Processing and Management* vol. 47, pp. 452–465, May, 2011.

Kang, I.-S., Na, S.-H., Lee, S., Jung, H., Kim, P., Sung, W.-K., and Lee, J.-H. On co-authorship for author disambiguation. *Information Processing & Management* 45 (1): 84–97, 2009.

Lapidot, I. Self-Organizing-Maps with BIC for Speaker Clustering. Tech. rep., IDIAP Research Institute, Martigny, Switzerland, 2002.

Levin, F. H. and Heuser, C. A. Evaluating the use of social networks in author name disambiguation in digital libraries. *Journal of Information and Data Management* 1 (2): 183–197, 2010.

Levin, M., Krawzyk, S., Bethard, S., and Jurafsky, D. Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology* 63 (5): 1030–1047, 2012.

Mitchell, T. M. *Machine Learning*. McGraw-Hill, New York, NY, USA, 1997.

On, B.-W., Elmacioglu, E., Lee, D., Kang, J., and Pei, J. Improving grouped-entity resolution using quasi-cliques. In *Proceedings of the 6th IEEE International Conference on Data Mining*. Hong Kong, China, pp. 1008–015, 2006.

On, B.-W. and Lee, D. Scalable name disambiguation using multi-level graph partition. In *Proceedings of the 7th SIAM International Conference on Data Mining*. Minneapolis, Minnesota, USA, pp. 575–580, 2007.

On, B.-W., Lee, D., Kang, J., and Mitra, P. Comparative study of name disambiguation problem using a scalable blocking-based framework. In *Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries*. Denver, CO, USA, pp. 344–353, 2005.

Pereira, D. A., Ribeiro-Neto, B. A., Ziviani, N., Laender, A. H. F., Gonçalves, M. A., and Ferreira, A. A. Using web information for author name disambiguation. In *Proceedings of the 2009 ACM/IEEE Joint Conference on Digital Libraries*. Austin, TX, USA, pp. 49–58, 2009.

Porter, M. F. An algorithm for suffix stripping. *Program* 14 (3): 130–137, 1980.

Rijsbergen, C. J. V. *Information Retrieval, 2nd edition*. Butterworths, London, 1979.

Shu, L., Long, B., and Meng, W. A latent topic model for complete entity resolution. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*. Shanghai, China, pp. 880–891, 2009.

Soler, J. M. Separating the articles of authors with the same name. *Scientometrics* 72 (2): 281–290, 2007.

Song, Y., Huang, J., Councill, I. G., Li, J., and Giles, C. L. Efficient topic-based unsupervised name disambiguation. In *Proceedings of the 7th ACM/IEEE Joint Conference on Digital Libraries*. Vancouver, BC, Canada, pp. 342–351, 2007.

Tang, J., Fong, A. C. M., Wang, B., and Zhang, J. A unified probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering* 24 (6): 975–987, 2012.

Torvik , V. I., Weeber, M., Swanson, D. R., and Smalheiser, N. R. A probabilistic similarity metric for Medline records: A model for author name disambiguation. *Journal of the American Society for Information Science and Technology* 56 (2): 140–158, 2005.

Torvik, V. I. and Smalheiser, N. R. Author name disambiguation in medline. *ACM Transactions on Knowledge Discovery from Data* 3 (3): 1–29, 2009.

Treeratpituk, P. and Giles, C. L. Disambiguating authors in academic publications using random forests. In *Proceedings of the 2009 ACM/IEEE Joint Conference on Digital Libraries*. Austin, TX, USA, pp. 39–48, 2009.

Veloso, A., Ferreira, A. A., Gonçalves, M. A., Laender, A. H., and Jr., W. M. Cost-effective on-demand associative author name disambiguation. *Information Processing & Management* 48 (4): 680–697, 2012.

Veloso, A., Meira Jr., W., and Zaki, M. J. Lazy associative classification. In *Proceedings of the International Conference on Data Mining*. Washington, DC, USA, pp. 645–654, 2006.

Wang, X., Tang, J., Cheng, H., and Yu, P. Adana: Active name disambiguation. In *Proceedings of the 11th International Conference on Data Mining*. Vancouver,Canada, pp. 794–803, 2011.

Yang, K.-H., Peng, H.-T., Jiang, J.-Y., Lee, H.-M., and Ho, J.-M. Author name disambiguation for citations using topic and web correlation. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*. Aarhus, Denmark, pp. 185–196, 2008.