Information Sciences 206 (2012) 42–62

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A tool for generating synthetic authorship records for evaluating author name disambiguation methods

Anderson A. Ferreira [a,b,*], Marcos André Gonçalves [a], Jussara M. Almeida [a], Alberto H.F. Laender [a], Adriano Veloso [a]

[a] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil
[b] Departamento de Computação, Universidade Federal de Ouro Preto, Brazil

ARTICLE INFO

Article history:
Received 14 May 2011
Received in revised form 19 February 2012
Accepted 8 April 2012
Available online 24 April 2012

Keywords:
Author name disambiguation
Digital library
Bibliographic citation
Synthetic generator

ABSTRACT

The author name disambiguation task has to deal with uncertainties related to the possible many-to-many correspondences between ambiguous names and unique authors. Despite the variety of *name disambiguation methods* available in the literature to solve the problem, most of them are rarely compared against each other. Moreover, they are often evaluated without considering a time evolving digital library, susceptible to dynamic (and therefore challenging) patterns such as the introduction of new authors and the change of researchers' interests over time. In order to facilitate the evaluation of name disambiguation methods in various realistic scenarios and under controlled conditions, in this article we propose SyGAR, a new Synthetic Generator of Authorship Records that generates citation records based on author profiles. SyGAR can be used to generate successive loads of citation records simulating a living digital library that evolves according to various publication patterns. We validate SyGAR by comparing the results produced by three representative name disambiguation methods on real as well as synthetically generated collections of citation records. We also demonstrate its applicability by evaluating those methods on a time evolving digital library collection generated with the tool, considering several dynamic and realistic scenarios.

1. Introduction

Scholarly digital libraries facilitate the organization, access, and retrieval of research results and articles. A common feature provided by most of these systems is to use author names to search for or navigate through sets of bibliographic citations.[1] However, an author name does not identify in a non-ambiguous way an underlying author, since there may be many-to-many correspondences between names and unique real-world persons. In order to solve the author name ambiguity problem, a disambiguation method must be used to correctly and unambiguously assign a citation record to one or more authors, already present in the digital library or not, despite the existence of multiple authors with the same or very similar names (polysems), or different name variations for the same author (homonyms) in the data repository.

The increasing necessity to automatically collect bibliographic data has motivated the research on author name disambiguation, and a rich variety of solutions have already been proposed based on supervised [17,23,27,31,49,48,50,52]

* Corresponding author at: Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil.
E-mail addresses: ferreira@dcc.ufmg.br (A.A. Ferreira), mgoncalv@dcc.ufmg.br (M.A. Gonçalves), jussara@dcc.ufmg.br (J.M. Almeida), laender@dcc.ufmg.br (A.H.F. Laender), adrianov@dcc.ufmg.br (A. Veloso).

[1] Here regarded as a set of bibliographic information such as author name, work title, publication venue title and publication year that are pertinent to a particular article.

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.04.022

or unsupervised [5,6,10,16,28,29,32,34,39–41,43,42,46,47,51,55] learning techniques. However, although most of these methods have been demonstrated to be relatively effective in a *static* scenario (in terms of error rate or similar metrics), none of them provides an ideal solution for the problem in the sense that they incorrectly disambiguate some authors. Yet, these methods are rarely compared against each other. More importantly, most previous evaluations do not consider a time evolving digital library, containing *dynamic* patterns such as the introduction of citations of new authors and the change of researchers' interests/expertises over time. Such patterns offer extra challenges to name disambiguation. Thus, previous evaluations leave open the very important question: *Would any of these methods effectively work on a dynamic and evolving scenario of a living digital library?*

One possible reason for the lack of previous evaluations under dynamic scenarios is that most previous analysis rely on a single or at most few (usually, no more than two) different real test collections, containing previously disambiguated authors. Moreover, the main existing collections, such as the one extracted from the DBLP digital library by Han et al. [27], do not contain temporal information, thus preventing any evaluation of evolving patterns. Furthermore, even if temporal information were available, an evaluation that relies only on a few collections is undoubtedly restricted to the scenarios captured in such collections.

A solid analysis of existing methods should consider various scenarios that do occur in real digital libraries. In addition to dynamic patterns, it should also address the robustness of existing methods under data errors, such as errors due to typographical errors and errors due to optical character recognition (OCR) and speech recognition. However, the construction of a real, previously disambiguated, temporal collection capturing different relevant dynamic scenarios and including various data errors is quite costly. An alternative is to build realistic *synthetic collections* that capture all scenarios of interest, under controlled conditions, while still inheriting the properties of real collections that are more relevant from the standpoint of existing name disambiguation methods. In particular, a generator of realistic synthetic collections, designed for the specific problem of name disambiguation, should be able to:

- Generate data whose disambiguation is non-trivial, following patterns similar to those found in real collections.
- Generate successive loads of data, at a certain frequency (e.g., one per year or month), containing new publications of the same set of authors, to assess the impact of the introduction of new publications into previously disambiguated digital libraries on the disambiguation methods.
- Generate data for new authors that were not originally included in the collection, simulating the situation in which the disambiguation method must identify the appearance of publications of authors not yet present in the digital library.
- Generate data reflecting changes in the authors' publication profiles (e.g., changes in the topics in which the authors publish), simulating changes of research interests over time.
- Introduce controlled errors on generated data, simulating errors caused by typos, misspelling, or OCR.

Therefore, in this article, we introduce and evaluate SyGAR, a new Synthetic Generator of Authorship Records, which addresses all the elicited requirements. SyGAR is capable of generating synthetic citation records given as input a list of disambiguated records of authorship citations extracted from a real digital library (input collection). The synthetically generated records follow the publication profiles of existing authors, extracted from the input collection. An author's profile is generated based on a set of distributions, namely, distribution of the number of coauthors per record, distribution of coauthor popularity, distribution of number of terms in a work title as well as distribution of topic (subject or interest) popularity of the given author. Each topic is associated with term and venue popularity distributions. SyGAR can be parameterized to generate records for new authors (not present in the input collection), for authors with dynamic profiles, as well as records containing typographical errors. For the best of our knowledge, SyGAR is the first generator of its kind, enabling and facilitating the investigation of several aspects of existing name disambiguation methods.

We validate SyGAR by comparing the results produced by three representative disambiguation methods on a standard real collection of (previously disambiguated) records and on synthetic collections produced using SyGAR parameterized with author profiles extracted from the real collection. The methods considered are: the supervised support vector machine-based method (SVM) proposed by Han et al. [27], the hierarchical heuristic-based method (HHC) proposed by Cota et al. [16] and the unsupervised k-way spectral clustering-based method (KWAY) proposed by Han et al. [29]. Our experiments show, for all three methods, a very good agreement in the performance obtained for real and synthetically generated collections, with relative differences of their performances under 10% in most cases.

To demonstrate the applicability of our generator, we evaluate the three aforementioned methods in three selected relevant real-world scenarios. In particular, we simulate a digital library evolving over a period of several years, during which (1) new publications of the same set of authors are introduced, (2) new authors with ambiguous names are introduced, at various rates, and (3) a fraction of the authors change their publication profiles. Our results indicate that the performance of SVM tends to degrade with time, particularly as new authors are introduced in the collection. In contrast, the performance of the unsupervised KWAY method, which uses privileged information regarding the number of authors in the digital library, tends to increase with time, except when there are changes in the authors' profiles. Overall, among the three methods, HHC has the best performance, which is due to its heuristic that was specially designed to address the name disambiguation task. In terms of their drawbacks, HHC suffers more with the addition of records of new authors, whereas SVM and KWAY are very sensitive to changes in the authors' profiles.

In sum, the main contribution of this article is the introduction of SyGAR, a new tested and validated synthetic generator of ambiguous groups of authorship records, which can help the evaluation, in several realistic scenarios and under controlled conditions, of solutions to the name disambiguation problem as well as to other problems related to name ambiguity. To demonstrate the applicability of our tool, we also present an evaluation of the relative performance of few representative disambiguation methods in relevant scenarios generated with SyGAR, that capture aspects of a live, real-world digital library over time.

This article is organized as follows. Section 2 discusses related work. Section 3 describes the design of SyGAR and its main components. Section 4 presents a brief overview of the three representative name disambiguation methods considered in our study. Section 5 presents our validation of SyGAR, whereas Section 6 demonstrates its applicability by presenting an evaluation of the selected disambiguation methods using synthetic citation records produced by SyGAR. Section 7 concludes the paper and offers possible directions for future work.

2. Related work

In this section, we present an overview of recent advances regarding the name ambiguity problem in the context of bibliographic citations. First, we briefly review representative automatic name disambiguation methods (Section 2.1). Then, we describe some previous synthetic data generators (Section 2.2) as well as existing collections used for evaluating author name disambiguation methods (Section 2.3).

2.1. Name disambiguation methods

The name disambiguation methods proposed in the literature adopt a wide spectrum of solutions [45] including approaches based on unsupervised techniques [5,6,10,16,28,29,32,34,39–41,43,42,46,47,51,55] and supervised ones [17,23,27,31,49,48,50,52].

Methods based on unsupervised techniques usually exploit similarities between citation records in order to group those records that are likely to be associated with the same author. For instance, in [6], Bhattacharya and Getoor propose a combined similarity function defined on attributes and relational information (i.e., coauthorship) and group records using a greedy agglomerative algorithm. In [16], the disambiguation method works in a two-step way in which coauthor names are first used to provide an initial grouping based on the purity of the clusters and then the work and publication venue titles are exploited in a hierarchical way to further group the citation records of a same author. An extension of this method that allows the name disambiguation task to be incrementally performed is presented in [10]. In [29], Han et al. use the cosine function to measure the similarity between two citation records and group the records using the K-way spectral clustering algorithm. In [46], a new probabilistic distance metric is proposed to group citation records of a same author. In [47], Song et al. use Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation to assign a vector of probabilistic topics to each citation and use this vector to group them. In [51], Velden et al. propose to disambiguate author names in scientific articles using coauthor names, self-citation and uncommon last names. Some unsupervised methods also adopt an iterative approach to identify the author of the records [5,28]. For instance, in [28], Han et al. use an unsupervised hierarchical version of naïve Bayes for modeling each author whereas in [5], Bhattacharya and Gettor exploit the Latent Dirichlet Allocation model assuming that each citation is constructed by choosing its authors given a distribution of author groups.

Supervised methods, that usually apply classification techniques [24], learn, from a training set containing pre-labeled examples, a "model" to either predict the author of a given citation record [23,27,52] or to determine if two citation records belong to the same author [17,31,48,50]. In [27], Han et al. use naïve Bayes and support vector machines to derive a function that infers the author of a citation record, while, in [23] and [52], Ferreira et al. present an associative [53] disambiguation method that exploits self-training to minimize the need for training data, being also capable of identifying new authors not present in the training set. In [17], Culotta et al. propose a supervised disambiguation algorithm which is both error-driven and rank-based. In [31], Huang et al. use DBSCAN [20] to cluster the citation records by author and online active selection support vector machines to learn a similarity metric to compare two records. In [48], Torvik et al. propose a probabilistic metric to determine the similarity among different MEDLINE records, and a heuristic to automatically generate the training set. Finally, in [50] Treeratpituk et al. use Random Forests [8] to learn a similarity function to compare pairs of citations.

Additionally, some other methods exploit data retrieved from the Web [32,34,42,43,55], topics [55] or data from specific sources, such as collections with information about all researchers of a country [18], as additional information for the disambiguation task. In a complementary direction, other studies exploit graph-based methods to disambiguate (author) names [21,39,41].

The name ambiguity problem also occurs in other contexts, such as the Web or email, in which distinct pieces of evidence can be explored. For instance, Diehl et al. use the email traffic network to disambiguate the names in the body of e-mail messages [19], while Galvez et al. exploit the use of finite-state transducers to identify variants of a person's name for disambiguating web search results [25]. Bekkerman et al. [4] use the link structure of the web pages and agglomerative/conglomerative double clustering for the same task. In [54], Vu et al. use web directories as a knowledge base to measure the similarity between two web pages containing ambiguous names. Yoshida et al. [56] exploit strong attributes (i.e., person, organization and place names) to make initial clusters of people-searching result pages and then use weak attributes (i.e., the

terms) and a bootstrapping algorithm to assign the pages to an existing or new cluster. There is also a workshop on web people searching, called WePS (standing for Web People Search) [1–3], that is concerned with organizing web results for a person's given name. Its two main tasks are clustering of results and attribute extraction. The clustering task aims at gathering together web pages about the same person in a same group while the attribute extraction task aims at extracting biographical attributes from each cluster for each person.

In sum, there is a multitude of different techniques in different contexts. Even in the specific context of scholarly digital libraries, our present focus, there are many techniques. However, these are rarely evaluated against each other, mainly in realistic and dynamic scenarios, due to previously discussed difficulties in constructing, for this task, collections that capture many real-life situations. Our goal here is to provide a tool that can facilitate future evaluations under realistic, dynamic and controlled scenarios. In the next section, we review some existing synthetic generators, which do not have all the capabilities of our tool, and a few already existing collections, which also have limitations regarding evaluation purposes.

2.2. Synthetic data generators

A variety of synthetic data generators is available in the literature, being most of them designed for a specific purpose. DSGen [12], for instance, is a tool to generate synthetic data representing personal information, such as first name, surname, address, dates, telephone and identifier numbers, which was developed as part of the Febrl deduplication system [13]. With that specific goal, DSGen generates synthetic data and duplicates them, inserting errors representing typographical errors. A more recent version of DSGen, introducing attribute dependencies as well as family and household data, is presented in [14].

In contrast, there are also a few general-purpose tools, such as DGL [9] and PSDG [30], which generate data based on specific languages used to describe several aspects of the data to be synthesized (e.g., distributions). These tools allow one to specify dependencies between attributes. However, neither of them can be parameterized with data from a given knowledge base, such as an existing real collection or a coauthorship graph. Such a feature is attractive as it can be exploited by the tool to infer attribute distributions from real-world data.

We are aware of only two synthetic data generators in the realm of digital libraries. The first one, SearchGen [38] generates synthetic workloads of scientific literature digital libraries and search engines. SearchGen was designed based on a characterization of the workload of CiteSeer,[2] extracted from usage logs. Li et al. validated the proposed tool by comparing the workload generated by SearchGen against logged workloads. SearchGen is fundamentally different from SyGAR, as our tool does not target the generation of workloads but rather of ambiguous citation records.

A tool that is more closely related to ours is the two-stage data generator proposed in [6]. The tool was designed to generate synthetic citations, specified by a list of authors. It works as follows. In the first stage, it builds a collaboration graph containing entities (i.e., authors) and relationships among them (i.e., coauthorships). In the second stage, it generates a collection of citations, each of which synthesized by first randomly selecting one entity and then randomly selecting its neighbors in the collaboration graph. SyGAR significantly differs from this tool. First, it generates values to other attributes, such as work and publication venue titles, in addition to author and coauthor names. Second, it is capable of generating a dynamically evolving collection, in which new authors, changes in an author's publication profiles and typographical errors may be introduced, at various rates. As such, our generator can be used to generate and simulate several controlled, yet realistic, long term scenarios, enabling an assessment of how distinct methods would behave under various conditions.

A preliminary version of SyGAR was discussed in [22]. In that prior version, SyGAR models an author's publication profile based on the distributions of the number of coauthors, coauthor popularity, number of terms in a work title, term popularity and venue popularity. By associating term and venue popularity distributions directly with the authors, our preliminary approach restricts the generation of citations containing only terms and venues that have been previously used by the authors. In its current version, SyGAR does not include term and venue popularity distributions as part of an author's profile. Instead, the profile of an author contains a distribution of *topics* (or research interests), and each topic has term and venue popularity distributions associated with it. This allows the generation of citations with work titles containing terms that have never been used by the authors or with a venue in which the authors have never published before. Moreover, the present tool allows one to generate data reflecting changes in the authors' publication profiles, simulating changes of research interests over time, and to introduce controlled errors on generated data, simulating errors caused by typos, misspelling, or OCR. Thus, the present tool is much more sophisticated and provides much more flexibility and richness to the process of generating synthetic citation records than its prior version.

2.3. Existing collections

In [27], Han et al. created a collection of citation records extracted from DBLP[3] which, with slight variations, has been used in several other studies [27,29,28,43,42,55]. Han et al. manually labeled the citation records using external sources of information such as authors' publication home pages, curriculum vitae, affiliations, and e-mail addresses. In case of doubt, they also sent emails to some authors to confirm the authorship of given citations. The records that had insufficient information to be checked

[2] http://citeseer.ist.psu.edu.
[3] http://dblp.uni-trier.de.

were eliminated from the collection. This collection totalizes 8442 citation records and 14 ambiguous groups with 480 authors. It contains the author names, work titles and publication venue titles of the citations.[4]

In [6], Bhattacharya and Getoor describe three collections, namely, CiteSeer, arXiv and BioBase. CiteSeer has only author names and work title attributes. It contains 1504 citation records of machine learning related publications with 2892 author names of 1165 authors. This collection was created by Aron Culotta and Andrew McCallum from the University of Massachusetts, Amherst. The arXiv collection, on turn, has 29,555 citation records of high energy physics related publications with 58,515 author names of 9200 authors. This collection, which has only the author name attributes, was used in KDD cup 2003.[5] Finally, the BioBase collection has 156,156 citation records of Elsevier publications on 'Immunology and Infectious Diseases' from 1998 to 2001 with 831,991 author names, of which 10,595 correspond to actual disambiguated names. All three collections are publicly available.[6]

In [16], Cota et al. describe a collection of citation records extracted from BDBComp, the Brazilian Computer Science Digital Library [35]. This collection contains citations for the authors sharing the top-10 most frequent author names, represented by their first name initials along with surnames. This collection was manually disambiguated using external sources of information such as the authors' publication home pages, affiliations, and e-mail addresses. It sums up 363 records associated with 184 distinct authors, approximately two records per author. Notice that, although this collection is smaller than the other ones, it is very difficult to disambiguate, because it carries many authors with very few records each[7].

Finally, in [33], Kang et al. describe the KISTI-AD-E-01-TestSet, a collection built by the Korea Institute of Science and Technology Information 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, an authorship record that contains the author name (i.e., the name of the author to be disambiguated), the coauthor names (i.e., the name of the rest of the authors) and other citation attributes (e.g., the work title, the publication venue title, the publication year) was built. To disambiguate this collection, the authors submitted a query composed of the surname of the author and the work title of each authorship record 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 6921 authors.[8]

Despite the existence of those collections, each of them corresponds to only a snapshot of the citation records extracted from the source digital library in a specific period, thus they are restricted to the scenarios captured during such periods. SyGAR, on the other hand, allows us to generate synthetic citation records for simulating many interesting scenarios, in a controlled way, over long periods of time. The design of our tool is described next.

3. SyGAR design

SyGAR is a tool for generating synthetic collections of citation records. Its design was driven by our goal of evaluating name disambiguation methods in more realistic, yet controlled, scenarios, with evolving digital libraries. It was thus designed to capture the aspects of real collections that are key to disambiguation methods and, therefore, to generate synthetic collections to evaluate them. These synthetic collections may be larger and span longer periods of time besides being representative of the real data with respect to *author publication profiles* (defined below).

SyGAR takes as input a real collection of previously disambiguated citation records, referred to as the *input collection*. Each such record is composed of the three attributes commonly exploited by disambiguation methods [16,23,27–29,37,42,43], namely, a list of author names and a list of unique terms present in the work title and the publication venue title. Authors with the same ambiguous name, and their corresponding records, are organized into *ambiguous groups* (e.g., all authors with name "C. Chen"). SyGAR also takes as input several other parameters, defined in Table 1 and described in the following sections, which are used in the data generation process.

As output, SyGAR produces a representative list of synthetically generated citation records, referred to as the corresponding *output collection*. Each generated record consists of the three aforementioned attributes. In particular, the (synthetic) work title is represented by a set of unique terms as opposed to a complete semantically-sound sentence, as most disambiguation methods typically exploit the former.

The overall generation process consists of three main steps, as shown in Fig. 1. SyGAR first summarizes the input collection into a set of attribute distributions that characterize the publication profiles of individual authors in the input collection. SyGAR builds publication profiles for all authors in the input collection, including those with ambiguous and non-ambiguous names. Next, the attribute distributions are used to generate synthetic records. Unless otherwise noted, only profiles of

[4] It is available at http://clgiles.ist.psu.edu/data/.

[5] http://www.cs.cornell.edu/projects/kddcup/index.html.

[6] The CiteSeer and the Arxiv collections can be found at http://www.cs.umd.edu/projects/linqs/projects/er/index.html while the BioBase dataset can be found at http://help.sciencedirect.com/robo/projects/sdhelp/about_biobase.htm.

[7] The BDBComp collection is available at http://www.lbd.dcc.ufmg.br/lbd/collections/disambiguation.

[8] It is available at http://www.kisti.re.kr.

Table 1
SyGAR input parameters.

Parameter	Description
N_{loads}	Number of loads to be synthesized
N_R	Total number of records to be generated per load
N_{Topics}	Number of topics
α_{Topic}	Threshold used to estimate distribution of topic popularity per citation (LDA model)
α_{Term}	Threshold used to estimate distribution of term popularity per topic (LDA model)
β_{Topic}	Minimum weight of topics that are associated to an author
$\alpha_{NewCoauthor}$	Probability of selecting a *new* coauthor
$\alpha_{NewVenue}$	Probability of selecting a *new* venue
$\%_{NewAuthors}$	Percentage of new authors to be generated in each load
$\%_{InheritedTopics}$	Percentage of topics to be inherited from a new author's main coauthor
$\%_{ProfileChanges}$	Percentage of authors that will have changes in their profiles in each load
δ	Shift parameter used to simulate changes in an author's profile
P^{FName}	Probability distribution of altering (removing, keeping or retaining) only the initial of the author's first name
P^{MName}	Probability distribution of altering (removing, keeping or retaining) only the initial of the author's middle name
$P^{LName}_{\#Mods}$	Probability distribution of the number of modifications in the author's last name
P^{LName}_{Mod}	Probabilities of inserting, deleting or changing one character or swapping two characters of the author's last name
$P^{Title}_{\#Mod}$	Probability distribution of the number of modifications in the work title
P^{Title}_{Mod}	Probabilities of inserting, deleting or changing one character or swapping two characters of the title
$P^{Venue}_{\#Mods}$	Probability distribution of the number of modifications in the venue
P^{Venue}_{Mod}	Probabilities of inserting, deleting or changing one character or swapping two characters of the venue

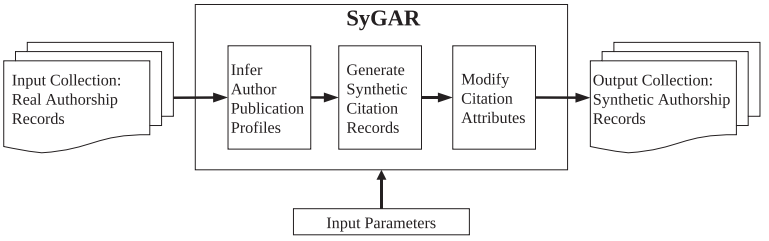


Fig. 1. SyGAR main components – SyGAR receives as input a disambiguated collection of citation records and builds publication profiles for all authors in the input collection. Then, the publication profiles are used to generate synthetic records. As a final step, SyGAR may introduce typographical errors in the output collection and change the citation attributes.

authors *with ambiguous names* are used to generate synthetic data.[9] As a final step, SyGAR changes the citation attributes, particularly the author names, so as to adhere to a pre-defined format (e.g., keep only the initial of the first name). In this step, it may also introduce typographical errors in the *output collection*. A detailed description of each step is presented in the following subsections.

3.1. Inferring publication profiles from the input collection

Each author's publication profile is characterized by her citation records. That is, the profile of author a is extracted from the input collection by summarizing her list of citation records into four probability distributions, namely:

1. a's distribution of number of coauthors per record – $P^a_{nCoauthors}$;
2. a's coauthor popularity distribution – $P^a_{Coauthor}$;
3. a's distribution of number of terms in a work title – P^a_{nTerms};
4. a's topic popularity distribution – P^a_{Topic}.

Each topic t is further characterized by two probability distributions:

1. t's term popularity distribution – P^t_{Term};
2. t's venue popularity distribution – P^t_{Venue}.

Finally, we also build a collection profile with:

[9] Profiles of authors with non-ambiguous names are used in the generation of profiles of new authors (Section 3.3), which relies on the profiles of all authors in the input collection.

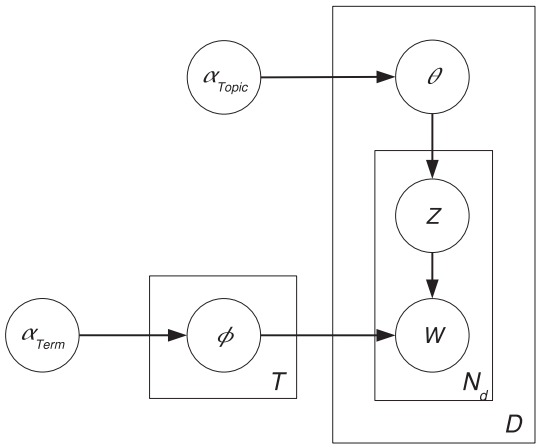


Fig. 2. A plate representation of the LDA [7] – the LDA model assumes that each citation record r follows the generative process. r draws the number of terms N_d in the work title according to a given distribution, draws a topic distribution θ according to a Dirichlet distribution model with parameter α_{Topic} and, for each term, chooses a topic z following the multinomial distribution θ and a term w from a multinomial probability conditioned on the selected topic z, given by distribution ϕ, which in turn is drawn according to a Dirichlet distribution with parameter α_{Term}.

1. probability distribution of the number of records per author with ambiguous names – $P^c_{nRecordsAuthors}$;
2. probability distribution of the number of records per author – $P^c_{nRecordsAllAuthors}$;
3. probability distribution of the number of records per ambiguous group – $P^c_{nRecordsGroup}$.

$P^a_{nCoauthors}$, $P^a_{Coauthor}$, P^a_{nTerms}, $P^c_{nRecordsAuthors}$ and $P^c_{nRecordsAllAuthors}$ can be directly extracted from the input collection by aggregating the citation records of each author a. We assume a's attribute distributions are statistically independent. In particular, we assume that, for any given citation, a's coauthors are independently chosen. Despite somewhat simplistic, these independence assumptions have also been made by most previous work in the context of name disambiguation [23,27–29,37]. More importantly, we show, in Section 5, that these assumptions have little (if any) impact on the performance of different disambiguation methods, as there is little difference in their results when applied to a real (input) collection and to synthetically generated (output) collections.

The main challenge here is to infer, from the input collection, the distributions of topic popularity for each author $\left(P^a_{Topic}\right)$, as well as the distributions of term and venue popularity associated with each topic (P^t_{Term} and P^t_{Venue}). Recall that the input collection does not contain any information on the topic(s) associated with each citation record. Thus, to address this challenge, SyGAR *models* each citation in the input collection as a finite mixture of a set of topics. In other words, each citation record r has an associated *topic distribution*, P^r_{Topic}.[10] Terms in the work title and publication venue title are drawn from corresponding distributions associated with the topics of the citation record, and not with the authors. This model is thus able to generate a citation record with a work title containing terms (or with a venue) not used yet by any of the authors, provided that such terms (or venue) are associated with a topic of their interests.

A first step to infer P^a_{Topic}, P^t_{Term} and P^t_{Venue} consists of deriving the distribution of topics for each citation record r in the input collection, P^r_{Topic}. This is performed using the Latent Dirichlet Allocation (LDA) generative model, previously proposed for modeling document contents [7]. LDA is a three-level hierarchical Bayesian model, as illustrated in Fig. 2. In this model, ϕ denotes a matrix of topic distributions, with a multinomial distribution of N_{Terms} terms for each of the N_{Topics} topics, which is drawn independently from a symmetric Dirichlet (α_{Term}) prior. N_{Terms} represents the total number of distinct terms in all work titles of the input collection whereas N_{Topics} is the total number of topics used to model the citations. Moreover, θ is the matrix of citation-specific weights for these N_{Topics} topics, each being drawn independently from a symmetric Dirichlet (α_{Topic}) prior. For each term, z denotes the topic responsible for generating that term, drawn from the θ distribution for that citation record, and w is the term itself, drawn from the topic distribution ϕ corresponding to z. In other words, the LDA model assumes that each citation record r follows the generative process described below:

1. Draw the number of terms $size_{Title}$ in the work title according to a given distribution, such as a Poisson distribution [7] or, in our case, the distribution of number of terms in a work title for a given author a, P^a_{nTerms};
2. Draw a topic distribution θ_r for citation record r according to a Dirichlet distribution model with parameter α_{Topic}; and
3. For each term i, $i = 1 \cdots size_{Title}$, choose a topic z_i following the multinomial distribution θ_r and a term w_i from a multinomial probability conditioned on the selected topic z_i, given by distribution ϕ_{z_i}, which in turn is drawn according to a Dirichlet distribution with parameter α_{Term}.

[10] $P^r_{Topic}(t)$ measures the strength at which a given topic t is related to the citation record r, normalized so as to keep the summation over all topics equal to 1. Thus, $P^r_{Topic}(t)$ can be seen as a *weight* associated with topic t for citation record r.

Thus, the LDA model has two sets of unknown parameters, namely, the topic distribution associated with each citation record r, θ_r, and the term popularity distribution of each topic j, ϕ_j, as well as the latent variables z corresponding to the assignments of individual terms to topics. Several strategies can be adopted to estimate θ_r and ϕ_j. As in [44,47], we use the Gibbs sampling algorithm [26]. This algorithm aims at generating a sequence of samples from the joint probability distribution of two or more random variables with the purpose of, for instance, estimating the marginal distributions of one of the variables. The Gibbs sampling algorithm constructs a Markov chain that converges to the posterior distribution of z by generating random samples from the observed data, and then uses the results to infer the marginal distributions θ_r and ϕ_j. The transitions between states of the Markov chain result from repeatedly drawing the topic of the ith term, z_i, from its distribution conditioned on all other variables, that is

$$P(z_i = j | w_i = m, z_{-i}, w_{-i}) \propto \frac{C^{WT}_{m_{-i}j} + \alpha_{Term}}{\sum_{m'} C^{WT}_{m'_{-i}j} + N_{Terms}\alpha_{Term}} \frac{C^{RT}_{r_{-i}j} + \alpha_{Topic}}{\sum_{j'} C^{RT}_{r_{-i}j'} + N_{Topics}\alpha_{Topic}} \quad (1)$$

In other words, it computes the probability that the topic assigned to the ith term (variable z_i) is j, given that the ith term (variable w_i) is m and given all topic assignments not including the one related to the ith term (z_{-i}). $C^{WT}_{m_{-i}j}$ is the number of times that term m is assigned to topic j excluding the current instance of term m, $C^{WT}_{m'_{-i}j}$ is the number of times that all terms in the collection are assigned to topic j excluding the current instance of m, $C^{RT}_{r_{-i}j}$ is the number of times that topic j is assigned to terms in citation record r excluding the current instance of m, $C^{RT}_{r_{-i}j'}$ is the number of times that all topics are assigned to terms in citation record r excluding the current instance of m.

From any sample from this Markov chain, we can estimate the probability of drawing a topic j for a citation r as

$$\theta_r(j) = \frac{C^{RT}_{rj} + \alpha_{Topic}}{\sum_{j'} C^{RT}_{rj'} + N_{Topics}\alpha_{Topic}} \quad (2)$$

and the probability of drawing a term m for a given topic j as

$$\phi_j(m) = \frac{C^{WT}_{mj} + \alpha_{Term}}{\sum_{m'} C^{WT}_{m'j} + N_{Terms}\alpha_{Term}} \quad (3)$$

These distributions correspond to the predictive distributions over new terms and new topics. According to Blei et al. [7], it is recommended to assign positive values to input parameters α_{Topic} and α_{Term} so as to allow the selection of new topics and new terms that have not been previously observed. In other words, positive values for these parameters ultimately imply in non-zero probabilities to all items (i.e., topics or terms) regardless of whether they have C^{RT}_{rj} (or C^{WT}_{mj}) equal to 0.

SyGAR follows the aforementioned procedure by processing all citation records in the input collection, one at a time. It uses the terms in the work titles to estimate the conditioned probability given by Eq. (1). After a number of iterations, it estimates the topic distribution of each citation record r, P^r_{Topic} (given by θ_r in Eq. (2)) and the term popularity distribution per topic t, P^t_{Term} (given by ϕ_j in Eq. (3)).

Afterwards, the tool infers the topic distribution P^a_{Topic} of each author a by combining the weights of the topics of all citation records in which a is an author. Only topics with weights greater than or equal to β_{Topic} (input parameter) are selected from each citation record of a, so as to avoid introducing topics of very little interest to a in P^a_{Topic}. SyGAR also infers the venue popularity distribution of each topic t, P^t_{Venue}, by combining the weights of t associated with citation records containing the same publication venue, provided that t has the largest weight among all topics of the given citation record, i.e., provided that t is the most related topic of the given citation record.[11]

Given the author profiles, SyGAR is ready to generate the synthetic citation records. It generates a number N_{loads} of batches of data representing a number of successive loads. For each load, it generates a number of records given by N_R or, alternatively, specified based on the distributions of the number of publications per author per load (as in Section 6.2).

Since SyGAR extracts publication profiles of all authors in the input collection, the term "author" was used up to this point in this Section to refer to any author in the input collection, regardless of whether she has an ambiguous name or not. Since our present goal is to evaluate disambiguation methods, we here use SyGAR to generate synthetic records only for authors *with ambiguous names*. Thus, for the sake of clarity, through the rest of this article and unless otherwise noted, we refer to authors *with ambiguous names*, the main target of our study, as simply *authors*, treating all other authors in the input collection as their *coauthors*.

The following three sections describe how SyGAR generates synthetic records for authors (with ambiguous names) already present in the input collection (Section 3.2) and for new authors (Section 3.3), as well as how it models dynamic publication profiles (Section 3.4) and how it modifies citation records in its final step (Section 3.5).

[11] These probabilities are combined by first summing up all values of $C^{RT}_{rj} + \alpha_{Topic}$ (numerator in Eq. (2)) for citations r and topics j of interest, and then normalizing them so as to keep the total probability equal to 1.

3.2. Generating records for existing authors

Each synthetic record for existing authors is created as follows:

1. Select one of the authors of the collection according to the desired distribution of number of records per author. Let it be a.
2. Select the number of coauthors according to $P^{a}_{nCoauthors}$. Let it be a_c.
3. Repeat a_c times:
   - with probability $1 - \alpha_{NewCoauthor}$, select one coauthor according to $P^{a}_{Coauthor}$;
   - otherwise, uniformly select a *new coauthor* among remaining coauthors in the input collection.
4. Combine the topic distributions of a and each of the selected coauthors. Let it be P^{all}_{Topic}.
5. Select the number of terms in the title according to P^{a}_{nTerms}. Let it be a_t.
6. Repeat a_t times: select one topic t according to P^{all}_{Topic} and select one term for the work title according to P^{t}_{Term}.
7. Select the publication venue:
   - with probability $1 - \alpha_{NewVenue}$, select a venue according to P^{t}_{Venue}, where t is the topic that was selected most often in Step 6;
   - otherwise, randomly select a *new venue* among remaining venues in the input collection.

Step 1 uses either the collection profile, i.e., $P^{c}_{nRecordsAuthors}$, or a distribution specified as part of the input. The latter may be specified by, for instance, providing the fractions of records to be generated for each author. This alternative input adds flexibility to our tool as it allows one to experiment with various scenarios by generating synthetic collections with varying numbers of records per author profile. Steps 2 and 5 use the distributions in the profile of the selected author. The same holds for Steps 3 and 7, although, with probabilities $\alpha_{NewCoauthor}$ and $\alpha_{NewVenue}$, SyGAR selects new coauthors and new venues (i.e., coauthors and venues that are not associated with the selected author in the input collection), respectively. We also note that, in Steps 3 and 6, we do not allow for a coauthor (or term) to be selected more than once.

The combined topic distribution P^{all}_{Topic} (Step 4) is obtained by first selecting only the topics that are shared by *all selected authors* (a and her coauthors). If there is no shared topic, we take the union of all topics associated with the selected author a and the coauthors. The combined distribution is built by, for each topic t, averaging $P^{a}_{Topic}(t)$ across all authors (a and the coauthors) and normalizing these values at the end so as to keep the summation over all topics equal to 1.

The seven steps are repeated a number of times equal to the target number of records in the new load.

3.3. Adding new authors

Another use for SyGAR is to generate records for large author populations, by building citation records not only for the authors present in the input collection but also for new (non-existing) authors. A variety of mechanisms could be designed to build such records. For the sake of demonstrating SyGAR's flexibility, we here adopt a strategy that exploits the publication profiles from author and co-authors, extracted from the input collection. Other (possibly more sophisticated) approaches will be designed in the future.

A new author a is created by first selecting one of its coauthors among all authors (with ambiguous and non-ambiguous names) in the input collection, i.e., using $P^{c}_{nRecordsAllAuthors}$. Let say it is c_a. The new author inherits c_a's profile, but the inherited topic and coauthor distributions are changed as follows. First, the new author inherits only a percentage $\%_{InheritedTopics}$ of the topics associated with c_a, i.e., the topics that are more strongly related to her (i.e., with largest weights). Let l_{Topic} be the list of inherited topics. The new author's topic popularity distribution is built by using the same weights c_a's distribution for the inherited topics, rescaling them afterwards so as to keep the summation equal to 1.

Similarly, we set a's coauthor list equal to c_a plus all coauthors of c_a that have at least one of the topics in l_{Topic} associated with them. Once again, the probabilities of selecting each coauthor are also inherited, and rescaled afterwards. However, we force that c_a appears in all records generated to the new author. This strategy mimics the case of a new author who, starting its publication career, follows part of the interests (topics) of one who will be a frequent coauthor (e.g., advisor or colleague).

Finally, the name of the new author is generated with the initial of the first name and the full last name of an existing author (i.e., an ambiguous author name), selected from the input collection using the distribution of the number of records per ambiguous group, i.e., $P^{c}_{nRecordsGroup}$.

Parameter $\%_{NewAuthors}$ specifies the percentage of new authors generated for each new load.

3.4. Changing an author's profile

SyGAR also allows one to experiment with dynamic author profiles, mimicking scenarios in which authors (with ambiguous names) may change their publication profiles over time due to shifts in interests, as occurs in real-world bibliographic digital libraries. Although SyGAR processes the input collection as a static snapshot of publication profiles, the tool can generate collections in which authors dynamically change their attribute distributions over successive loads.

In the lack of a previous characterization of dynamic properties of author publication, SyGAR currently implements a simple strategy to change the *topic distribution* of an author a, illustrated in Fig. 3a. It first sorts the topics associated with a according to their probabilities (i.e., P^{a}_{Topic}) so as to have a histogram as close to a bell shape as possible (i.e., mode in the cen-

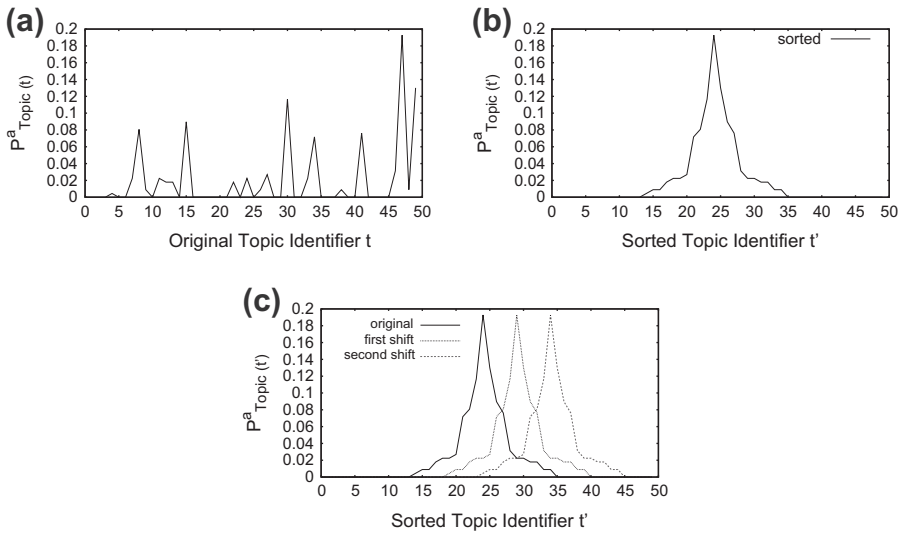


Fig. 3. Changing author *a*'s profile by altering her topic distribution. (a) The original topic distribution of author *a*. (b) The topics associated with *a* sorted according to their probabilities $\left(P^a_{Topic}\right)$ so as to have a histogram as close to a bell shape as possible. (c) The topic distribution shifted along the *x*-axis by a factor $\delta = 5$; 2 shifts are shown in the figure.

ter and least probable topics in both extremes), as illustrated in Fig. 3b. It then shifts the distribution along the *x*-axis by a factor of δ, at each load. Fig. 3c illustrates four successive changes in an author's profile using δ equals to 5.

By carefully choosing δ, this procedure guarantees that changes occur as softly as desired, mimicking the case of an author smoothly increasing/decreasing her interest in some topics over time.

Parameter $\%_{ProfileChanges}$ specifies the percentage of authors that will experience changes in their profiles in each load.

3.5. Modifying citation attributes

The final step in the citation record generation process consists of modifying the citation attributes according to several input probability distributions (see Table 1). Two mandatory changes refer to how an author's first and middle names should be presented in the citation record. There are three possibilities: completely remove the first/middle name, keep the first/middle name entirely and keep only the initial of the first/middle name. Probability distributions P^{FName} and P^{MName} are used to make the selections, which are applied to the names of all authors and coauthors in the synthetic citations.

Next, six input distributions may be used to introduce typographical errors in the generated records. $P^{LName}_{\#Mods}$, $P^{Title}_{\#Mods}$ and $P^{Venue}_{\#Mods}$ are used to draw the number of modifications in each author's last name, work title and publication venue, respectively, whereas P^{LName}_{Mod}, P^{Title}_{Mod} and P^{Venue}_{Mod} are used to draw the type of each such modification in each attribute. Four modifications are possible, namely, insert, remove or change one character and swap two randomly selected characters.

In its current version, SyGAR allows for the easy experimentation with a multitude of relevant scenarios (see examples in Section 6) that occur in real digital libraries. We intend, in the future, to design even more sophisticated mechanisms to add new authors to the output collection as well as new strategies to introduce changes in the profiles of existing authors and in the synthetic records.

As a final note, we emphasize that, although SyGAR was designed to help addressing the name disambiguation task, it can be used to generate any collection of citation records, as long as a real collection is provided as source of author profiles. Thus, we believe it can be used to study other problems related to bibliographic digital libraries as well (e.g., scalability issues). SyGAR is implemented in Java and will be available for public use in due time.

4. Representative name disambiguation methods

Before evaluating SyGAR, we first describe the name disambiguation methods we used to validate and illustrate the applicability of our tool. As the methods available in the literature adopt a variety of solutions, including unsupervised and supervised techniques (see Section 2), we here select three methods, each one being representative of a different technique. Next, we briefly introduce the selected methods.

4.1. SVM-based method

The SVM-based name disambiguation method, here referred to as simply SVM, was proposed by Han et al. [27]. The authors associate each author name with an author class and train the classifier for that class. Each citation record is

represented by a feature vector with elements within its attribute values (e.g., terms of the title, coauthor names, etc.) as features, and frequencies of these elements as feature weights.

Support vector machines (SVMs) [15] receive training vectors $x_i \in R^n$ (input space), $i = 1,2,\ldots,l$, and a vector $y \in \{-1,1\}^l$, and find an optimal decision hyperplane that minimizes the risk of erroneous classifications. SVMs produce a model that predicts target values of data instances in the testing set. The model is produced using a training set, which has feature vectors representing the instances that will be classified along with their class label. SVMs require both positive and negative examples to learn how to classify the data. It has two main parameters, namely, the type of kernel function k_f, which indicates the structure of the solution function, and cost j, which controls the penalty given to classification errors in the training process.

4.2. Unsupervised heuristic-based hierarchical method

Cota et al. [16] proposed a heuristic-based hierarchical clustering method (HHC) for name disambiguation that involves two steps. In the first step, the method creates clusters of citation records with similar author names that share at least a similar coauthor name. Then, in the second step, the method successively fuses clusters of records with similar author names based on the similarity of two other citation attributes, work title and publication venue title.

More specifically, two clusters are fused if they include citation records with similar work titles or similar publication venue titles, which is identified according to two input parameters (work and publication venue similarity thresholds). In each fusion, the information of fused clusters is aggregated (i.e., all words in the titles are grouped together) providing more information for the next round of fusion. This process is successively repeated until no more fusions are possible.

To calculate the similarity between two clusters, Cota et al. [16] used each word in the work or publication venue title as a term and calculated the cosine between two clusters using feature vectors, where each feature corresponds to the product of *TF* (term frequency) and *IDF* (inverse document frequency) (i.e., $TF*IDF$) of its corresponding term.

4.3. K-way spectral clustering-based method

This unsupervised method, proposed by Han et al. [29], follows a clustering approach based on the K-way spectral clustering technique [57]. The method, here referred to as simply KWAY, does not use any training data, but it does assume that the number of correct clusters is previously known and, therefore, it is a parameter of the method. In other words, although some input is needed from the user, there is no explicit learning phase. Thus, the KWAY method can be considered unsupervised. Moreover, since it is based on a state-of-the-art clustering technique that uses privileged information (i.e., the number of correct clusters), which may not be realistic for the problem, it produces results that may be considered upper-bounds for clustering solutions.

In the KWAY disambiguation method, each citation record is mapped into a vertex of an undirected graph, and the edge weight between two vertices represents the similarity between the corresponding citation records. The disambiguation problem consists of splitting the graph so that citation records that are more similar to each other belong to the same cluster. Each citation record is modeled as a feature vector, with each feature corresponding to an element (e.g., a word) of a given instance of each attribute (e.g., title, author, etc.) of the citation record. The authors considered two options for defining the feature weights, namely, TFIDF, the product $TF * IDF$, and Normalized Term Frequency, *NTF*. *NTF* is given by $ntf(i,d) = freq(i,d)/maxfreq(d)$, where $freq(i,d)$ refers to the frequency of feature i in citation record d and $max\ freq(d)$ refers to the maximal frequency of any feature in record d. We here use the TFIDF scheme as it produced better results in the original paper.

5. SyGAR's validation

We validate SyGAR by comparing the performance of the selected name disambiguation methods on real and synthetically generated collections as well as by comparing attribute distributions (author/coauthor and topic distributions) of both collections. We start by describing the real collection (Section 5.1) and the performance metric used in our evaluation (Section 5.2). Validation results are presented in Section 5.3.

5.1. Real collection

The real collection of citation records used in our study is a subset of the citation records extracted by Han et al. from DBLP [27]. It contains a total of 4287 records associated with 220 distinct authors (with ambiguous names), with an average of approximately 20 records per author. The collection contains, in total, 11 ambiguous groups. The number of records and the number of distinct authors with the same ambiguous name within each group are presented in Table 2. Slight variations of the original collection [27] have also been used in several other studies [16,23,27,29,28,43,42,55].

Table 2
Real collection – an extract of DBLP.

Ambiguous group	Number of records	Number of authors
A. Gupta	576	26
A. Kumar	243	14
C. Chen	798	60
D. Johnson	368	15
J. Martin	112	16
J. Robinson	171	12
J. Smith	921	29
K. Tanaka	280	10
M. Brown	153	13
M. Jones	260	13
M. Miller	405	12

5.2. Performance metric

The ultimate goal of a disambiguation method is to separate all records within each ambiguous group into a number of subgroups (i.e., clusters), one for each different (disambiguated) author name. In order to evaluate the performance of the disambiguation methods, we here use the K metric [36], which captures a balance between two specific clustering metrics, namely, average cluster purity (ACP) and average author purity (AAP). K, ACP and AAP are applied separately to each ambiguous group.

ACP evaluates the purity of the generated clusters with respect to manually generated reference clusters, i.e., whether the generated clusters include only records belonging to the reference clusters. In our case, the reference clusters are the result of the manual disambiguation previously performed on each group. Thus, ACP is equal to 1 if the generated clusters are pure. ACP is computed as

$$\mathrm{ACP} = \frac{1}{N}\sum_{i=1}^{q}\sum_{j=1}^{R}\frac{n_{ij}^2}{n_i}$$

where R is the number of reference clusters (manually generated clusters); N is the total number of records in the ambiguous group; q is the number of clusters automatically generated by the disambiguation method; n_{ij} is the total number of records of the automatically generated cluster i that belong to the reference cluster j; and n_i is the total number of records of the automatically generated cluster i.

AAP, in turn, evaluates the fragmentation of the automatically generated clusters with respect to the reference clusters. Its values vary between 0 and 1. The smaller the number of fragmented clusters, the closer it is to 1. AAP is computed as

$$\mathrm{AAP} = \frac{1}{N}\sum_{j=1}^{R}\sum_{i=1}^{q}\frac{n_{ij}^2}{n_j}$$

where n_j is the total number of records in the reference cluster j.

The K metric consists of the geometric mean of ACP and AAP, thus capturing both purity and fragmentation of the clusters generated by each method. The K metric is given by

$$\mathrm{K} = \sqrt{\mathrm{ACP} \times \mathrm{AAP}}$$

5.3. Validation results

Since our main goal is to use SyGAR to evaluate representative disambiguation methods, our main validation consists of assessing whether the synthetically generated collection captures the aspects of the corresponding real collection (and its ambiguous groups) that are of relevance to the disambiguation methods. Towards that goal, we compare the K result obtained when each of the three selected disambiguation methods is applied to the real collection and its corresponding synthetic versions.

In our validation experiments, we set $\alpha_{Topic} = \alpha_{Term} = 0.00001$, thus allowing, with a very small probability, the selection of any topic/term, regardless of whether they were associated with the selected authors/topics in the input collection (see Section 3.1). We believe this leads to the generation of more realistic synthetic collections. Moreover, we set $\beta_{Topic} = 0.07$, i.e., to infer the topic distribution of each author, we combine the topics with weights greater than or equal to 0.07 in each citation record of such author, avoiding introducing topics with very little interest to her in the topic distribution. The number of authors and the number of records per author in the synthetic collections are set to be the same as in the input collection, as both parameters have impact on the effectiveness of the methods, and thus should be kept fixed for validation purposes. In other words, we let N_R and $P^c_{nRecordsAuthors}$ be the same as in the input collection and make $N_{loads} = 1$. For validation purposes,

Table 3
SyGAR validation – average K results and 95% confidence intervals for real and synthetically generated collections (N_{Topics} = 300). Statistical ties are in bold.

Collection	KWAY	SVM	HHC
Real	0.589 ± 0.006	0.799 ± 0.006	**0.770** ± 0.006
Synthetic 1	0.470 ± 0.008	0.698 ± 0.005	**0.753** ± 0.013
Synthetic 2	0.486 ± 0.005	0.704 ± 0.005	0.750 ± 0.011
Synthetic 3	0.479 ± 0.006	0.711 ± 0.005	0.752 ± 0.005
Synthetic 4	0.503 ± 0.010	0.714 ± 0.006	0.755 ± 0.006
Synthetic 5	0.477 ± 0.004	0.710 ± 0.006	0.751 ± 0.011

we set $\%_{NewAuthors} = \%_{InheritedTopics} = \%_{ProfileChanges} = \alpha_{NewCoauthor} = \alpha_{NewVenue} = \delta = 0$, keep first and middle names of each author as in the input collection and avoid introducing any typographical error in the synthetic collections. We experiment with N_{Topics} equal to 300 and 600. We further discuss issues related to the sensitivity of SyGAR to these parameters later in this section.

Regarding the parameters for the methods, for SVM, we used the implementation provided by the LibSVM package [11], with RBF (Radial Basis Function) as the kernel function, where the best γ and cost values were obtained from the training data using the *Grid* program, available with the LibSVM package. For KWAY, we used the implementation of the K-way spectral clustering provided by the University of Washington spectral clustering working group[12] and the number of authors in the collections as the target number of clusters to be generated. For HHC, we used the same values specified in [16] for the work and venue title similarity thresholds.

For the sake of evaluation, we divided the real collection as well as each synthetic collection generated from it into two equal-sized portions, by randomly splitting the author records into two parts. One is the training data and the other is the test set. We then applied each method to each ambiguous group in the test set. The supervised method uses the training data to learn the disambiguation model. We repeated this process 10 times for each collection, presenting results that are averages of the 10 runs.

Table 3 shows average K results, along with corresponding 95% confidence intervals, for the three disambiguation methods applied to the real collection and to five synthetically generated collections[13] using N_{Topics} = 300. Note that the synthetic collections are only slightly more difficult to disambiguate than the real one. Indeed, K results for KWAY, SVM and HHC methods are, on average, around only 17%, 11% and 2.3%, respectively, smaller in the synthetic collections, including a statistical tie between the real and a synthetic collection using the HHC method (marked in bold). We notice that, the number of distinct terms in the work titles used by each author in the synthetic collection with N_{Topics} = 300 is around 9% greater than in the real collection. Since KWAY relies directly on the similarity among the records to group them, which uses the work title, this may explain the larger difference for this method. HHC, on the other hand, first groups by coauthor and only uses the information in the work and publication titles for minimizing the fragmentation problem, while SVM, relies on the training data, being more robust to these changes. This may explain the smaller differences between these methods when applied to the synthetic and real collections.

We consider these results very good, given the complexity of the data generation process, and considering that SyGAR allows for the selection of title terms and venues not previously associated with an author. In other words, the synthetic collections, built using SyGAR, are mimicking reasonably well the real data.

Table 4 shows similar results for synthetic collections built using N_{Topics} = 600. Note that these collections are easier to disambiguate and the K results are closer to those produced for the real collection. Indeed, comparing real and synthetic collections, results for KWAY and SVM are, on average, only 9% and 4% smaller in the synthetic collections, whereas the HHC results are slightly better in the synthetic collections (3.3%, on average). These results further show that SyGAR is capable of capturing the aspects of the real collection that are relevant to the disambiguation methods.

The reason why using 600 topics instead of 300 leads to synthetic collections on which the disambiguation methods produce results closer (or even slightly better) to the results for the corresponding real collections may be explained as follows. As the number of topics increases, the number of authors sharing any given topic tends to decrease. As a consequence, when building a synthetic citation, there is a higher chance that a term selected for a given topic (Eq. (3)) has been actually used, in the real collection, by the author to which that topic was associated. Recall that, when generating a citation, if the selected authors share no topic, SyGAR combines all topics of individual authors. This happens with the majority of the citations generated when we set N_{Topics} = 600. Thus, in this case, the chance of generating synthetic citations with terms that were used by at least one of the authors in the real collection is higher, which ultimately makes the synthetic citations look more similar to the real ones, at least with respect to title terms. This leads to synthetic collections that better resemble the real ones, therefore justifying the similar performance of the methods.

To better understand the sensitivity of SyGAR to some of its key parameters, we evaluate the results of two of the selected methods, namely SVM and HHC, when applied to synthetic collections generated using various values of α_{Topic}, α_{Term}, β_{Topic}, and N_{Topics}. We report, for each method, the relative difference of its performance on the real and synthetic collections, here

[12] http://www.stat.washington.edu/spectral.
[13] These collections were built based on the same input parameters, differing only with respect to the seed used in the random number generator.

Table 4
SyGAR validation – average K results and 95% confidence intervals for real and five synthetically generated collections (N_{Topics} = 600).

Collection	KWAY	SVM	HHC
Real	0.589 ± 0.006	0.799 ± 0.006	0.770 ± 0.006
Synthetic 1	0.550 ± 0.004	0.781 ± 0.007	0.792 ± 0.008
Synthetic 2	0.536 ± 0.011	0.751 ± 0.006	0.790 ± 0.009
Synthetic 3	0.531 ± 0.007	0.761 ± 0.007	0.799 ± 0.012
Synthetic 4	0.531 ± 0.009	0.774 ± 0.005	0.796 ± 0.006
Synthetic 5	0.511 ± 0.010	0.747 ± 0.006	0.801 ± 0.008

referred to as the *relative error*. A positive error implies that the synthetic collection is harder to disambiguate than the real one. We report average results of five runs, omitting confidence intervals for the sake of clarity.

We start by showing, in Fig. 4a and c, average errors for SVM and HHC, respectively, as we vary α_{Topic} and α_{Term} from 10^{-6} to 10^{-2} (setting both to the same value in each case), and N_{Topics} from 100 to 600, while keeping β_{Topic} fixed at 0.07. We note that, as both α_{Topic} and α_{Term} increase, the gap between the results on synthetic and real collections tends to increase significantly for both methods, particularly for large number of topics. The synthetic collections become harder to disambiguate for larger values of α_{Topic} and α_{Term}. This is because larger values of both parameters impact the computation of P^r_{Topic} and P^t_{Term} (Eqs. 2 and 3, respectively) more significantly. This is particularly true if N_{Topics} is large, since counters C^{RT}_{rj} and C^{WT}_{mj}, inferred from the input collection, tend to decrease as the number of topics increases. In other words, larger values of α_{Topic} and α_{Term} may introduce too much noise in the estimates of P^r_{Topic} and P^t_{Term}, ultimately generating synthetic collections that are much harder to disambiguate than the real collection. The same can be noticed, though to a less extent, for smaller number of topics.

Moreover, the errors also tend to decrease as the number of topics (N_{Topics}) increases, provided that the values of α_{Topic} and α_{Term} are not very large. As previously discussed, the larger the number of topics, the higher the chance of generating citations with terms that were used by at least one of its authors in the real collection. One extreme case is N_{topics} = 600 and $\alpha_{Topic} = \alpha_{Term} = 10^{-5}$, when, as previously discussed, this happens to most generated citations and both methods produce

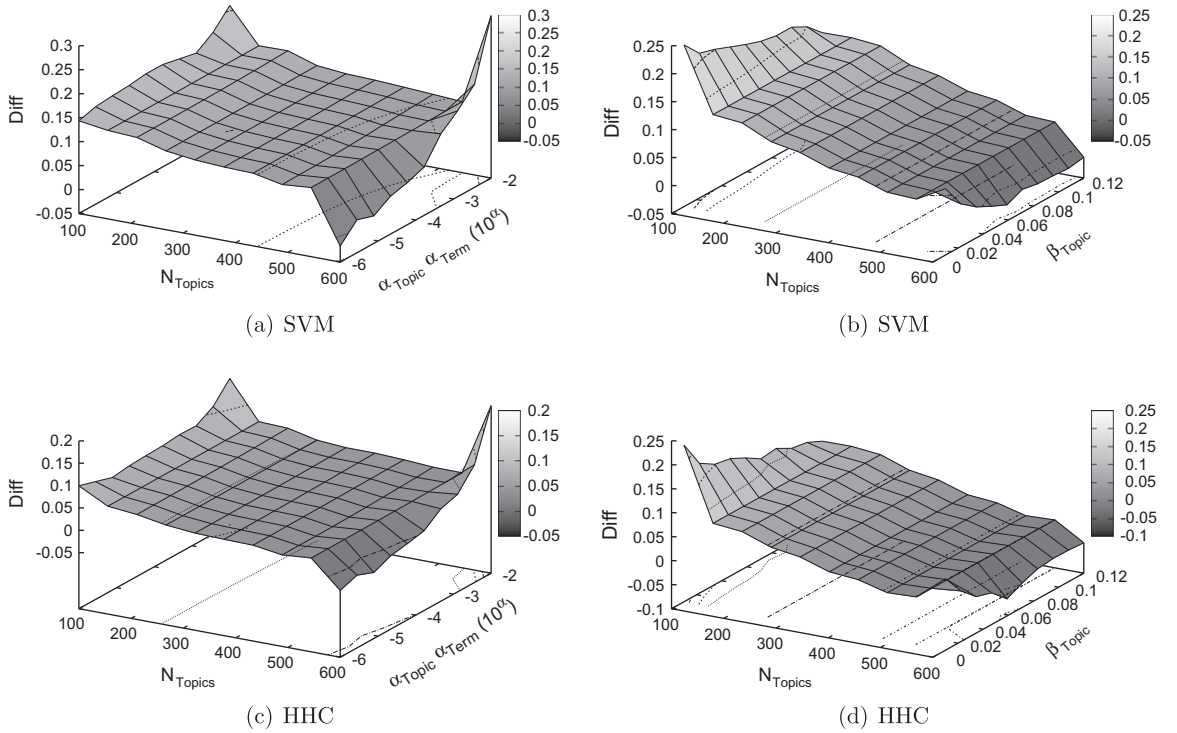
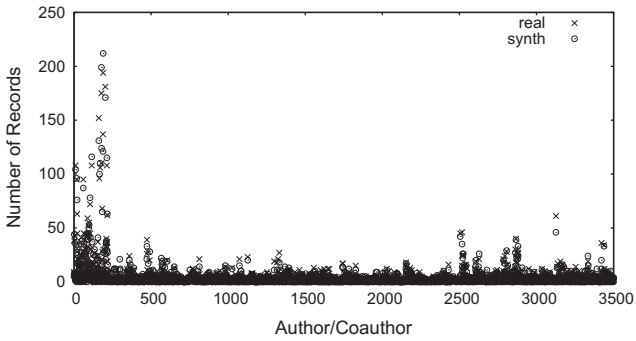
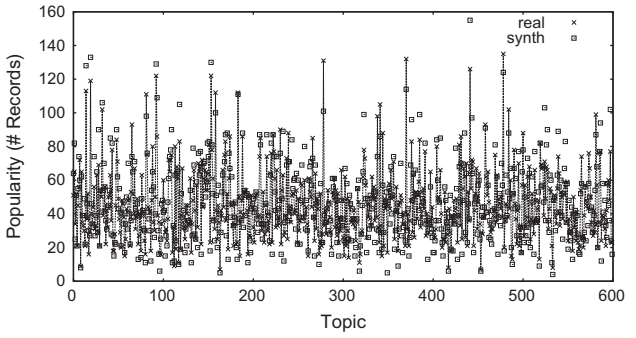


Fig. 4. Sensitivity of SyGAR to α_{Topic}, α_{Term}, β_{Topic} and N_{Topics} – relative error between performance of each method on synthetic and real collections. (a and c) Show the results of SVM and HHC, respectively, when applied to synthetically generated collections using various values of α_{Topic}, α_{Term} and N_{Topics}, keeping β_{Topic} = 0.07. (b and d) Show the results of SVM and HHC, respectively, when applied to synthetically generated collections using various of β_{Topic} and N_{Topic}, keeping $\alpha_{Topic} = \alpha_{Term} = 10^{-5}$.



(a) Number of Records per Author/Coauthor



(b) Topic Popularity (N_{Topics}=600)

Fig. 5. SyGAR validation. We use $\alpha_{Topic} = \alpha_{Term} = 10^{-5}$ and $\beta_{Topic} = 0.7$.

results that are very close to those obtained with the real collection. Thus, we suggest to use $\alpha_{Term} = 10^{-5}$, $\alpha_{Topic} = 10^{-5}$ and $N_{Topics} = 600$.

Next, Fig. 4b and d show average errors for SVM and HHC, respectively, as we vary β_{Topic} from 0.01 to 0.12 and N_{Topics} from 100 to 600, keeping $\alpha_{Topic} = \alpha_{Term} = 10^{-5}$. In general, both methods tend to produce results closer to those obtained with the real collection for larger values of β_{Topic}. This is expected as β_{Topic} represents the minimum weight of topics that can be associated with an author. Thus, in general, larger values of β_{Topic} tend to reduce the chance of associating to an author a topic that is of little interest to her. So, the β_{Topic} value must be lower or equal to 0.10. Therefore, we suggest to set $\beta_{Topic} = 0.10$.

We further validate SyGAR by comparing some of the attribute distributions in the real and synthetic collections. As a sanity check, Fig. 5a shows the distributions of the number of records per author/coauthor ($P^{c}_{nRecordsAllAuthors}$) in the real and in a synthetically generated collection. Clearly, both distributions are very similar, as expected. Fig. 5b, in turn, shows the popularity (in terms of number of citations) of topics, in the real and in a synthetic collection built using $N_{Topics} = 600$. Recall that this metric is *not* directly manipulated by SyGAR. Once again, the curves show very similar patterns. Similar agreement was also obtained for collections generated using other values of N_{Topics} as well as for other attribute distributions.

6. Evaluating disambiguation methods with SyGAR

We demonstrate the applicability of SyGAR by evaluating the SVM, KWAY and HHC disambiguation methods in three realistic scenarios generated by our tool. We start by describing these scenarios in Section 6.1. We then present our experimental setup in Section 6.2 and discuss the most relevant results from our evaluation in Section 6.3. We emphasize that our goal here is *not* to thoroughly evaluate the selected methods but rather to show how our tool can be used to evaluate existing methods in relevant realistic situations.

6.1. Analyzed scenarios

We envision three scenarios that capture some relevant dynamic patterns observed in real digital libraries. All three scenarios encompass a live digital library (DL) evolving over the period of several years. In its initial state, the DL is a collection of synthetic citations. At the end of each year, a load is performed into the DL with new citations of existing, and, possibly, of new ambiguous authors, depending on the specific scenario. We choose to model yearly loads, using as parameters the

average yearly publication rates of authors in each ambiguous group, extracted from DBLP (see Section 6.2). However, the load period could be easily changed.

Scenario 1 consists of an evolving digital library with new citations introduced at each new load, assuming a fixed author population with static publication profiles. In other words, Scenario 1 captures solely the impact of an evolving DL. Only authors (with ambiguous names) in the original input collection are considered and they *do not* change their profiles during successive loads, keeping their topic and coauthor distributions as extracted from the input collection.

Scenario 2 considers the introduction of new authors to the existing author population. New authors are added to the collection at a given rate in each successive load. As described in Section 3.3, a new author inherits a percentage of the topics of an author that will be considered as her main coauthor (e.g., an advisor). Moreover, all publications of a new author have her main coauthor in the author list.

Finally, *Scenario 3* considers authors with dynamic profiles. A percentage of the current authors make small changes in their profiles before each new load, i.e., their topic distributions are shifted by a factor δ, as explained in Section 3.4. The changes are very small, but are performed at a constant rate over the years. Although this might not be very realistic, it allows us to test the limits of the disambiguation methods under dynamic publication profiles. As we are unaware of previous studies measuring profile change rates in real-word digital libraries, any choice of rate would be arbitrary.

Thus, the envisioned scenarios allow us to evaluate the robustness of the selected disambiguation methods to three key real-world aspects: (1) the evolution of the DL, (2) the inclusion of new authors with ambiguous names into the DL and (3) changes in author profiles. We emphasize that these are only a few of the scenarios that can be generated using SyGAR. For instance, scenarios with different, possibly heterogeneous, profile change rates, i.e., different values of δ for different authors, can also be devised, being the loads easily produced by SyGAR. Building and experimenting with other scenarios is subject of future work.

6.2. Experimental setup

We performed experiments with the same collection used in Section 5, containing 11 ambiguous groups, as shown in Table 2. The parameters for the disambiguation methods were also the same as described in Section 5. For each scenario, the number of synthetic citation records in the initial state s_0 of the digital library is the same as in the real collection. Ten successive data loads, one per year, are generated using SyGAR (i.e., $N_{loads} = 10$), parameterized by the real collection as source of publication profiles as well as with additional inputs according to the specific scenario.

Starting at state s_i, the new citation records generated by SyGAR are disambiguated using each one of the three methods and the results are incorporated into the corresponding DL version, which evolves into state s_{i+1}. If the supervised SVM method is used, SyGAR is also used to generate a training set containing the same number of citations of the DL at its initial state s_0. This training set is used by SVM to "learn" its model to disambiguate the records generated at each load. For both KWAY and HHC methods, the generated records are first incorporated into the current state of the DL and the disambiguation is performed with all records.

For each new load, SyGAR generates records for authors already in the DL and, in Scenario 2, for new authors. The synthetic citations are generated using $N_{Topics} = 600$, $\beta_{Topic} = 0.10$ and $\alpha_{Topic} = \alpha_{Term} = 10^{-5}$. Moreover, in all three scenarios, we set $\alpha_{NewVenue} = \alpha_{NewCoauthor} = 0$, thus restricting the selection of venues and coauthors for an author's new citation to those already associated with her in the input collection.

We also format author and coauthor names according to probabilities p that match the observed patterns in the input collection. In particular, we retain either only the initial of the first name ($p = 0.53$) or the complete first name ($p = 0.47$). Moreover, regarding the middle name, we either keep only the initial ($p = 0.37$), remove it ($p = 0.53$) or keep it completely ($p = 0.10$). Finally, we introduce no typographical errors in any attribute.

For experiments with Scenario 2, the number of new authors to be added at each new load is specified as a fraction $\%_{NewAuthors}$ of the total number of authors in the DL at its current state. We experiment with values of $\%_{NewAuthors}$ equal to 5% and 10%. Each new author inherits 80% of the topics associated with her most frequent coauthor ($\%_{InheritedTopics} = 80\%$). We note that newly added authors remain as part of the DL throughout the rest of the experiment, i.e., records are generated for these authors in all successive loads.

Moreover, for experiments with Scenario 3, changes are introduced in a percentage $\%_{ProfileChanges}$ of author profiles across successive loads using a shift $\delta = 5$. We experimented with $\%_{ProfileChanges}$ equal to 10%, 50% and 100%. In this case, in each yearly load a different set of authors from the previous state is chosen to have their profiles changed.

Table 5
Distribution of average number of publications per year per author (DBLP: 1984–2008).

	Average number of publications per year			
	One (%)	Two (%)	Three (%)	>Four (%)
New authors	55	30	10	5
Existing authors	14	42	28	16

Finally, the distribution of the number of records generated for each author is built from the data presented in Table 5, which shows the distribution of the average number of publications per year per (existing and new) author. These distributions were extracted from DBLP, counting the number of publications of each author of three selected ambiguous groups during the period of 1984–2008. We selected groups "C. Chen", "A. Gupta" and "D. Johnson" which, as shown in Table 2, have very different author population sizes. "C. Chen" is a very large ambiguous group with 60 different authors. "D. Johnson", on the contrary, is much smaller, and "A. Gupta" has an intermediary number of authors.

For loads s_1 to s_{10}, the generation of new records use the distributions shown in Table 5. We chose to use that distribution because Han et al.'s DBLP collection, which we use here, did not have temporal information, so the number of records per author $(P^c_{nRecordsAuthors})$ is a cumulative measure, and using it would certainly generate distortions depending on the length of the career of that author. For generating the successive loads, the yearly rates of publication are more important.

6.3. Evaluation of results

The following subsections present our evaluation of the three selected methods in each considered scenario built using SyGAR. Our evaluation is carried out by computing the K value at each state of the DL. The results reported in the following sections are averages of five runs. Corresponding 95% confidence intervals are usually very tight, indicating errors on the reported means that fall below 12% in all cases.

6.3.1. Scenario 1: evolving DL with static author population and profiles

Fig. 6 shows, for each disambiguation method, the average K value computed over all 11 ambiguous groups in each state of the digital library over the 10-year period. Corresponding 95% confidence intervals are also shown. Note that the relative order of the methods, in terms of achieved performance, remains the same through all states: HHC outperforms SVM, which, in turn, outperforms KWAY. However, the three methods have very different behaviors as new loads of citations are introduced into the DL.

SVM's performance, for example, tends to decrease over time: while it starts in the first load (s_0) with an average K value equals to 0.78, these values fall to levels around 0.66 in the successive loads. Indeed after 10 loads, SVM's performance degrades by 15%. This degradation is possibly due to errors caused by imprecise models learned for authors with very few records in the training set. These errors are cumulative in the successive loads, calling for a retraining of SVM. Analyzing SVM with retraining is not an easy task as factors such as errors introduced in the collection may affect the results of these experiments. Thus, we leave it for future work.

KWAY, on the contrary, experiences an increasing improvement in effectiveness as new citations are added. This occurs because there is incrementally more information about each author, helping KWAY to better characterize them. Indeed, the gain in performance after 10 loads reaches 32%. Unlike both SVM and KWAY, HHC remains with approximately the same performance, varying by at most 2%, throughout all 10 successive loads. This is possibly due to the specific heuristics exploited by HHC for the name disambiguation task (see [16] for details), in contrast to the general purpose techniques used by SVM and KWAY.

As consequence of such distinct behaviors, we find that, while in the beginning (i.e., state s_0) HHC outperforms SVM by only 2% (on average) and SVM outperforms KWAY by 60% (on average), corresponding performance gains switch to 18% and only 3%, respectively, after the last load of new citations.

6.3.2. Scenario 2: introduction of new authors

We now use SyGAR to analyze the impact on each method of introducing new authors to the current author population. Fig. 7 shows average K values and corresponding 95% confidence intervals for each method on collections built using $\%_{NewAuthors}$ equal to 5% and 10%, and $\%_{InheritedTopics}$ equal to 80%.

The behaviors of both KWAY and SVM follow trends very similar to those observed in Fig. 6: whereas SVM suffers performance degradation, KWAY actually improves in performance as new loads of citations are added to the DL. However, we

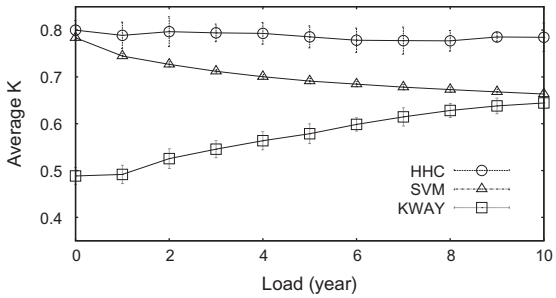


Fig. 6. Scenario 1 – evolving DL with static author population and publication profiles.

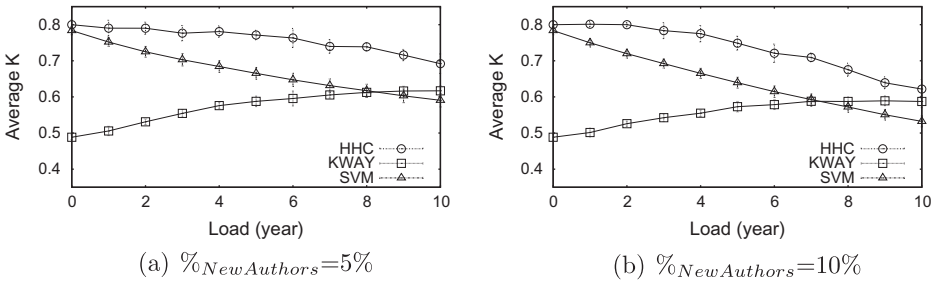


Fig. 7. Scenario 2 – evolving DL and addition of new authors ($\%_{InheritedTopics}$ = 80%).

note a clear detrimental impact of the introduction of new authors on both methods. SVM's performance degrades much faster for $\%_{NewAuthors}$ = 10% than for $\%_{NewAuthors}$ = 5%. Indeed, in comparison with the case of static author population (Fig. 6), the average K values after the last load is 20% and 11% worse for $\%_{NewAuthors}$ equal to 10% and 5%, respectively. Recall that SVM uses the same training set, containing only records of the existing authors in state s_0, to disambiguate the DL in all states. Therefore, SVM is unable to recognize new authors, thus introducing errors into the DL when disambiguating their records. Once again, SVM requires retraining when facing the addition of new authors to the DL, a subject of future study.

Similarly, the improvement in performance experienced by KWAY becomes less significant as the fraction of new authors introduced at each load increases. This happens because of the increase in the number of authors, which implies in higher ambiguity and a higher inherent difficulty in distinguishing them. In comparison with the case reported in Fig. 6, KWAY's performance after the last load is 9% and 4% worse for $\%_{NewAuthors}$ equal to 10% and 5%, respectively. In fact, for both values of $\%_{NewAuthors}$, KWAY outperforms SVM after the last load.

Fig. 7 also shows that, like SVM and KWAY, HHC also suffers a significant performance degradation with the introduction of new authors. Indeed, for $\%_{NewAuthors}$ = 10%, the difference in average performance between HHC and KWAY drops from 64% to only 6% after the last load. In comparison with the case of static author population, average K values after the last load are 21% and 12% worse for $\%_{NewAuthors}$ equal to 10% and 5%, respectively.

6.3.3. Scenario 3: dynamic author profiles

Finally, Fig. 8a–c show average K values and corresponding 95% confidence intervals when a fraction $\%_{ProfileChanges}$ equal to 10%, 50% and 100% of the authors experience changes in their profiles at each new load. All three methods greatly suffer if facing dynamic changes in profiles. KWAY, in particular, which experiences performance improvements in both Scenarios 1 and 2, now suffers some degradation for values of $\%_{ProfileChanges}$ greater than or equal to 50%. In particular, taking Scenario 1 and the performance of each method after the last load as basis for comparison, we note that SVM's performance degrades by 16%, 31% and 34% for $\%_{ProfileChanges}$ equal to 10%, 50% and 100%, respectively. HHC, in turn, experiences a performance degra-

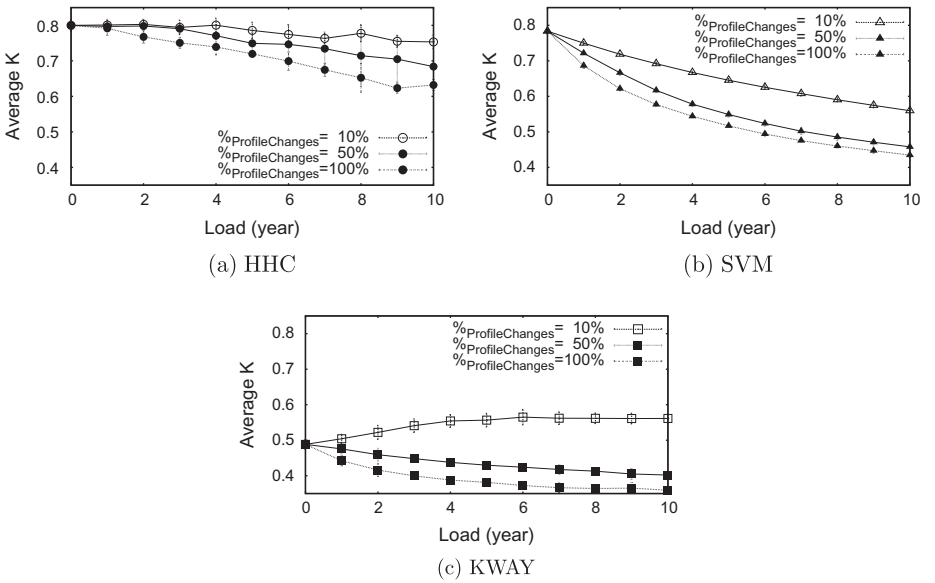


Fig. 8. Scenario 3 – dynamic author profiles (δ = 5 and $\%_{ProfileChanges}$ = 10%, 50% and 100%).

dation of 4%, 13% and 19% in the respective cases, being therefore more robust than SVM in this scenario. KWAY, which seems very robust to Scenarios 1 and 2, still experiences some performance improvement (by as much as 14%) if $\%_{ProfileChanges}$ is equal to only 10%. Notice however, that this improvement is smaller than in the scenarios in which we did not have profile changes (in that case, improvements went up to 32%). However, for values of $\%_{ProfileChanges}$ equal to 50% and 100%, its performance degrades by 18% and 26%, respectively. While KWAY was able to take advantage of the increase in information in Scenarios 1 and 2, the change in the profile of existing authors confounds this method.

In sum, the performance of SVM tends to degrade over time, particularly as new authors are introduced in the collection. In contrast, the performance of the unsupervised KWAY method, which uses privileged information regarding the number of authors in the digital library, tends to increase with time, except when there are changes in the author profiles. Overall, among the three methods, the heuristic-based method HHC, designed specifically to address the name disambiguation problem, has the best performance in the analyzed situations.

7. Conclusions and future work

In this article we presented SyGAR, a new tool that generates synthetic citation records given a collection of (previously disambiguated) real citation records. The synthetically generated records follow the publication profiles of existing authors, extracted from the input collection. Moreover, SyGAR allows for the simulation of several real-world scenarios such as the introduction of new authors (not present in the input collection), dynamic changes in an author's publication profile as well as the introduction of typographical errors in the synthetic citations.

A preliminary and very simple version of SyGAR was discussed in [22]. In that prior version, SyGAR models an author's publication profile based on the distributions of the number of coauthors, coauthor popularity, number of terms in a work title, term popularity and venue popularity. In its current version, the profile of an author contains a distribution of *topics* (or research interests), and each topic has term and venue popularity distributions associated with it. This allows the generation of citations with work titles containing terms that have never been used by the authors or with a venue in which the authors have never published before. Moreover, the present tool allows one to generate data reflecting changes in the authors' publication profiles, simulating changes of research interests over time, and to introduce controlled errors on generated data, simulating errors caused by typos, misspelling, or OCR.

SyGAR was designed with the goal of supporting the evaluation of name disambiguation methods in various realistic scenarios. Thus, we validate it by comparing the results produced by three representative disambiguation methods on a standard real collection and on synthetic collections produced using our tool. The selected methods are the supervised SVM-based method [27], the heuristic HHC method [16] and the unsupervised KWAY clustering-based method [29]. Our validation experiments show a very good agreement in the performance obtained for all three methods for real and synthetically generated collections.

We further analyzed SyGAR by demonstrating its applicability to evaluate the selected methods under three real-world scenarios, namely, the evolution of a DL with static author population and publication profiles, the introduction of new authors and the dynamic changes in the author's profiles. Although not the focus of this article, results obtained with this initial evaluation are very interesting, demonstrating the potential of the analysis that can be performed with collections built by SyGAR. For instance, these results indicate that the performance of SVM tends to degrade with time, particularly as new authors are introduced in the collection. In contrast, the performance of the unsupervised KWAY method, which uses privileged information regarding the number of authors in the digital library, tends to increase with time, except when there are changes in the author's profiles. Overall, among the three methods, the heuristic HHC method, designed specifically to address the name disambiguation task, has the best performance in most analyzed scenarios.

Future work includes the evaluation of many other scenarios exploring the capabilities of SyGAR. Further, we also intend to improve the tool with more sophisticated strategies to add new authors to the digital library and to dynamically change the authors' publication profiles.

Acknowledgments

This research is partially funded by InWeb – The National Institute of Science and Technology for the Web (MCT/CNPq/FAPEMIG Grant No. 573871/2008-6), and by the authors's individual research grants from CAPES, CNPq, and FAPEMIG.

References

[1] J. Artiles, A. Borthwick, J. Gonzalo, S. Sekine, E. Amigó, WePS-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks, in: CLEF (Notebook Papers/LABs/Workshops), Padua, Italy, 2010, pp. 1–15.
[2] J. Artiles, J. Gonzalo, S. Sekine, The SemEval-2007 WePS evaluation: establishing a benchmark for the web people search task, in: Proceedings of the 4th International Workshop on Semantic Evaluations, Prague, Czech Republic, 2007, pp. 64–69.
[3] J. Artiles, J. Gonzalo, S. Sekine, WePS 2 evaluation campaign: overview of the web people search clustering task, in: Proceedings of the 2nd Web People Search Evaluation Workshop, 18th WWW Conference, Madrid, Spain, 2009.
[4] R. Bekkerman, A. McCallum, Disambiguating web appearances of people in a social network, in: Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, 2005, pp. 463–470.
[5] I. Bhattacharya, L. Getoor, A latent dirichlet model for unsupervised entity resolution, in: Proceedings of the Sixth SIAM International Conference on Data Mining, Bethesda, MD, USA, 2006, pp. 47–58.

[6] I. Bhattacharya, L. Getoor, Collective entity resolution in relational data, ACM Transactions on Knowledge Discovery from Data 1 (2007) 5:1–5:36.
[7] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[8] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.
[9] N. Bruno, S. Chaudhuri, Flexible database generators, in: Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 2005, pp. 1097–1107.
[10] A.P. Carvalho, A.A. Ferreira, A.H.F. Laender, M.A. Gonçalves, Incremental unsupervised name disambiguation in cleaned digital libraries, Journal of Information and Data Management 2 (2011) 289–304.
[11] C.-C. Chang, C.-J. Lin, LibSVM: A Library for Support Vector Machines, 2001. Software: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
[12] P. Christen, Probabilistic data generation for deduplication and data linkage, in: Proceedings of the 6th International Conference on Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, vol. 3578, Springer, Brisbane, Australia, 2005, pp. 109–116.
[13] P. Christen, Febrl – an open source data cleaning, deduplication and record linkage system with a graphical user interface, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Las Vegas, Nevada, USA, 2008, pp. 1065–1068.
[14] P. Christen, A. Pudjijono, Accurate synthetic generation of realistic personal information, in: Proceedings of the 13th Pacific–Asia Conference on Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 5476, Springer, Bangkok, Thailand, 2009, pp. 507–514.
[15] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.
[16] R.G. Cota, A.A. Ferreira, M.A. Gonçalves, A.H.F. Laender, C. Nascimento, An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations, Journal of the American Society for Information Science and Technology 61 (2010) 1853–1870.
[17] A. Culotta, P. Kanani, R. Hall, M. Wick, A. McCallum, Author disambiguation using error-driven machine learning with a ranking loss function, in: Sixth International Workshop on Information Integration on the Web, Vancouver, Canada, 2007, pp. 32–37.
[18] C.A. D'Angelo, C. Giuffrida, G. Abramo, A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments, Journal of the American Society for Information Science and Technology 62 (2011) 257–269.
[19] C.P. Diehl, L. Getoor, G. Namata, Name reference resolution in organizational email archives, in: Proceedings of the Sixth SIAM International Conference on Data Mining, Bethesda, MD, USA, 2006, pp. 70–81.
[20] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, 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, 1996, pp. 226–231.
[21] X. Fan, J. Wang, X. Pu, L. Zhou, B. Lv, On graph-based name disambiguation, ACM Journal of Data and Information Quality 2 (2011) 10:1–10:23.
[22] A.A. Ferreira, M.A. Gonçalves, J.M. Almeida, A.H.F. Laender, A. Veloso, SyGAR – A synthetic data generator for evaluating name disambiguation methods, in: Proceedings of the 13th European Conference on Digital Libraries, Corfu, Greece, 2009, pp. 437–441.
[23] A.A. Ferreira, A. Veloso, M.A. Gonçalves, A.H.F. Laender, 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, 2010, pp. 39–48.
[24] D. Fisch, B. Khbeck, B. Sick, S.J. Ovaska, So near and yet so far: new insight into properties of some well-known classifier paradigms, Information Sciences 180 (2010) 3381–3401.
[25] C. Galvez, F. de Moya Anegón, Approximate personal name-matching through finite-state graphs, Journal of the American Society for Information Science and Technology 58 (2007) 1960–1976.
[26] T. Griffiths, M. Steyvers, Finding scientific topics, Proceedings of the National Academy of Sciences 101 (2004) 5228–5235.
[27] H. Han, C.L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis, Two supervised learning approaches for name disambiguation in author citations, in: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, Tuscon, AZ, USA, 2004, pp. 296–305.
[28] H. Han, W. Xu, H. Zha, C.L. Giles, 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, 2005a, pp. 1065–1069.
[29] H. Han, H. Zha, C.L. Giles, Name disambiguation in author citations using a k-way spectral clustering method, in: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, Denver, CO, USA, 2005b, pp. 334–343.
[30] J.E. Hoag, C.W. Thompson, A Parallel General-Purpose Synthetic Data Generator, SIGMOD Record, vol. 36, 2007, pp. 19–24.
[31] J. Huang, S. Ertekin, C.L. Giles, 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, 2006, pp. 536–544.
[32] P. Kanani, A. McCallum, 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, 2007, pp. 429–434.
[33] I.-S. Kang, P. Kim, S. Lee, H. Jung, B.-J. You, Construction of a large-scale test set for author disambiguation, Information Processing & Management 47 (2011) 452–465.
[34] I.-S. Kang, S.-H. Na, S. Lee, H. Jung, P. Kim, W.-K. Sung, J.-H. Lee, On co-authorship for author disambiguation, Information Processing & Management 45 (2009) 84–97.
[35] A.H.F. Laender, M.A. Gonçalves, P.A. Roberto, BDBComp: building a digital library for the brazilian computer science community, in: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, Tuscon, AZ, USA, 2004, pp. 23–24.
[36] I. Lapidot, Self-Organizing-Paps with BIC for Speaker Clustering, Technical Report IDIAP Research Institute Martigny, Switzerland, 2002.
[37] D. Lee, B.-W. On, J. Kang, S. Park, Effective and scalable solutions for mixed and split citation problems in digital libraries, in: Proceedings of the 2nd International Workshop on Information Quality in Information Systems, Baltimore, Maryland, 2005, pp. 69–76.
[38] H. Li, W.-C. Lee, A. Sivasubramaniam, C.L. Giles, SearchGen: A synthetic workload generator for scientific literature digital libraries and search engines, in: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital Libraries, Vancouver, BC, Canada, 2007, pp. 137–146.
[39] B. Malin, Unsupervised name disambiguation via social network similarity, in: Proceedings of the Workshop on Link Analysis, Counterterrorism, and Security, at the SIAM International Conference on Data Mining, Newport Beach, CA, 2005, pp. 93–102.
[40] B.-W. On, E. Elmacioglu, D. Lee, J. Kang, J. Pei, An effective approach to entity resolution problem using quasi-clique and its application to digital libraries, in: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC, USA, 2006, pp. 51–52.
[41] B.-W. On, D. Lee, Scalable name disambiguation using multi-level graph partition, in: Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, Minnesota, USA, 2007, pp. 575–580.
[42] D.A. Pereira, B. Ribeiro-Neto, N. Ziviani, A.H.F. Laender, M.A. Gonçalves, A generic web-based entity resolution framework, Journal of the American Society for Information Science and Technology 62 (2011) 919–932.
[43] D.A. Pereira, B.A. Ribeiro-Neto, N. Ziviani, A.H.F. Laender, M.A. Gonçalves, Ferreira, A.A., Using web information for author name disambiguation, in: Proceedings of the 2009 Joint International Conference on Digital Libraries, Austin, TX, USA, 2009, pp. 49–58.
[44] M. Rosen-Zvi, T.L. Griffiths, M. Steyvers, P. Smyth, The author-topic model for authors and documents, in: Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, 2004, pp. 487–494.
[45] N.R. Smalheiser, V.I. Torvik, Author name disambiguation, Annual Review of Information Science and Technology 43 (2009) 287–313.
[46] J.M. Soler, Separating the articles of authors with the same name, Scientometrics 72 (2007) 281–290.
[47] Y. Song, J. Huang, I.G. Councill, J. Li, C.L. Giles, Efficient topic-based unsupervised name disambiguation, in: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, Vancouver, BC, Canada, 2007, pp. 342–351.
[48] V.I. Torvik, N.R. Smalheiser, Author name disambiguation in MEDLINE, ACM Transactions on Knowledge Discovery from Data 3 (2009) 11:1–11:29.
[49] V.I. Torvik, M. Weeber, D.R. Swanson, N.R. Smalheiser, A probabilistic similarity metric for Medline records: a model for author name disambiguation, Journal of the American Society for Information Science and Technology 56 (2005) 140–158.
[50] P. Treeratpituk, C.L. Giles, Disambiguating authors in academic publications using random forests, in: Proceedings of the 9th ACM/IEEE-CS Joint International Conference on Digital Libraries, Austin, TX, USA, 2009, pp. 39–48.

[51] T.A. Velden, A.-u. Haque, C. Lagoze, Resolving author name homonymy to improve resolution of structures in co-author networks, in: Proceeding of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries, Ottawa, Ontario, Canada, 2011, pp. 241–250.
[52] A. Veloso, A. Ferreira, M. Gonçalves, A. Laender, W. Meira Jr., Cost-effective on-demand associative author name disambiguation. Information Processing & Management, in press. http://dx.doi.org/10.1016/J.IPM.2011.08.005.
[53] A. Veloso, W. Meira Jr., M.A. Gonçalves, H.M. de Almeida, M.J. Zaki, Calibrated lazy associative classification, Information Sciences 181 (2011) 2656–2670.
[54] Q.M. Vu, T. Masada, A. Takasu, J. Adachi, Using a knowledge base to disambiguate personal name in web search results, in: Proceedings of the 2007 ACM Symposium on Applied Computing, Seoul, Korea, 2007, pp. 839–843.
[55] K.-H. Yang, H.-T. Peng, J.-Y. Jiang, H.-M. Lee, J.-M. Ho, Author name disambiguation for citations using topic and web correlation, in: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries, Aarhus, Denmark, 2008, pp. 185–196.
[56] M. Yoshida, M. Ikeda, S. Ono, I. Sato, H. Nakagawa, Person name disambiguation by bootstrapping, in: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 2010, pp. 10–17.
[57] H. Zha, X. He, C.H.Q. Ding, M. Gu, H.D. Simon, Spectral relaxation for K-means clustering, in: Neural Information Processing Systems, MIT Press, 2001, pp. 1057–1064.