A Real-time Heuristic-based Unsupervised Method for Name Disambiguation in Digital Libraries

Muhammad Imran
Qatar Computing Research Institute
Tornado Tower, West Bay
Doha, Qatar
mimran@qf.org.qa

Syed Zeeshan Haider Gillani
University of Trento
Via Sommarive 5
Trento, Italy
gillani@disi.unitn.it

Maurizio Marchese
University of Trento
Via Sommarive 5
Trento, Italy
marchese@disi.unitn.it

ABSTRACT

This paper addresses the problem of name disambiguation in the context of digital libraries that administer bibliographic citations. The problem occurs when multiple authors share a common name or when multiple name variations for an author appear in citation records. Name disambiguation is not a trivial task, and most digital libraries do not provide an efficient way to accurately identify the citation records for an author. Furthermore, lack of complete meta-data information in digital libraries hinders the development of a generic algorithm that can be applicable to any dataset. We propose a heuristic-based, unsupervised and adaptive method that also examines users' interactions in order to include users' feedback in the disambiguation process. Moreover, the method exploits important features associated with author and citation records, such as co-authors, affiliation, publication title, venue, etc., creating a multilayered hierarchical clustering algorithm which transforms itself according to the available information, and forms clusters of unambiguous records. Our experiments on a set of researchers' names considered to be highly ambiguous produced high precision and recall results, and decisively affirmed the viability of our algorithm.

Keywords

Digital libraries, Name disambiguation, Bibliographic data

1. INTRODUCTION

Digital libraries (DLs) such as DBLP, Google Scholar, and Microsoft Academia conserve and provide bibliographic citation data, thus allowing the search and discovery of relevant publications (i.e., citation records) in a centralized way. Access to citation records of an author requires querying a DL with a full or partial name, and retrieving a list of citation records that match the given input. The name disambiguation problem occurs when multiple authors share a common name, or an author's multiple name variations appear in DLs due to name abbreviations or misspellings. In both scenarios, it can be difficult to be certain about the accuracy of retrieved records. For example, the full version of an abbreviated author name "M. Imran" may refer to "Muhammad Imran" or "Malik Imran", two completely different persons but in DLs both are referred to as "M. Imran". This phenomenon of citation merger, the case of two persons having the same name variations (e.g., "M. Imran") is known as "mixed citation" [12]. The same type of confusion can exist for an author named "Muhammad Imran", which is a full name shared by many other authors. This problem results in splitting an author's citation records among others whose names are exactly the same, and is characterized as "split citation" [12].

In recent years, many attempts have been made to solve the problem. Primarily two kinds of approaches have been proposed so far: supervised approach [2, 3, 4, 9, 8, 19], and unsupervised approach [12, 20, 10, 18, 17, 11]. The former requires a set of training data to train a classifier for further disambiguation processing, a process that involves tedious human effort, as manual labeling of citation records is not a trivial and straightforward task for many reasons. The latter approach works without the training of a classifier, thus considering only the importance of an author's or a citation's meta-data. Despite many attempts, both DLs and community efforts have failed to solve the ambiguity problem and provide an adaptive and generic solution that is extensible in behavior.

In this paper we propose a heuristic-based, unsupervised and adaptive method that stretches to multi-layer design with real-time execution. The proposed method exploits both author-specific meta-data features (e.g., author name, co-authors, affiliation, etc.) as well as citation record-specific meta-data features (e.g., publication title, published venue, etc.). The method also considers users' feedback during the process, to tune subsequent steps for better performance. As not all DL's provide a complete set of meta-data features, our algorithm uses an adaptive layering approach, and uses the available features for a particular DL to measure overall weight for clustering. Furthermore, as our algorithm is based on an unsupervised approach, it does not require training a classifier with a set of correct manually labeled data. In summary, the main contributions of this work are:

JCDL-WOSP '2013 Indianapolis, USA
July 2013

1. Providing a sensible step-by-step disambiguation ap-

proach which is generic and adaptive in nature, and can be applied to any DL.

2. Providing an end-to-end solution for the name disambiguation problem, starting from the retrieval of citation records from a DL (i.e., DBLP, RKBexplorer), pre-processing, incorporating users' feedback and ultimately disambiguating the names. Our disambiguation service works as a wrapper on top of DLs. That means that, given a query to a particular DL, it first retrieves citation records, and then disambiguates them for further analysis.

3. Fabricating an unsupervised multi-layered hierarchal clustering method, by employing author-specific as well as publication-specific features to disambiguate authors at each layer, without any manual labeling of data.

4. Presenting a comparative experiment using our algorithm and a well known approach (i.e., ADANA [20]). The results demonstrate that our algorithm outperforms that approach in precision, recall and F1 measures, and proves the viability of our approach.

2. MOTIVATIONAL SCENARIOS

The problem of getting mixed citations and split citations (as defined above) in users' search results occurs across DLs, and even though most DLs claim to have precise and accurate data, they are unable to deal with the name ambiguity problem with complete precision. The examples below were collected from various DLs, including [1], ACM[2] and Google Scholar[3].

A real example of Mixed Citations is the name "Bogdan Alexe" which is the name of two different authors; one who is affiliated with "USC" working on schema mappings and the other working with "ETH Zurich" on computer vision. This name yields 15 mixed citation records in ACM DL and 20 in DBLP DL.

Similarly, a real example of Split Citations is "Robert Schreiber", which corresponds to two different authors and additionally to two different variations as well (R. Schreiber and Robert Schreiber). One of them is affiliated with "Hewlett Packard" and with topics of imaging and vision, and the other is affiliated with "NASA Ames Research Center", working on distributed large systems. They have combined mixed publications totaling 48 in "DBLP DL" and 39 in "ACM DL".

These two examples are representative of the magnitude of the mixed/split citation problem which motivates our work. Additional motivation comes from the innumerable cases of Chinese authors who have similar surnames, specifically DBLP DL, which is loaded with such cases. Moreover, DBLP and other libraries are frequently not informative enough in their meta-data, and lack pieces of important information such as an author's affiliation, publication dates, etc. The severity of the problem can be assessed from the recent contest (KDD cup 2013[4]) published by Microsoft Academic for implementing a system of name disambiguation.

Author's Name	No. of publications	No. of people
Michael Smith	38	24
Philip J. Smith	33	3
Yang Yu	72	20
Qiang Shen	70	3
Michael Lang	24	24
Hui Yu	32	22
Charles Smith	7	4
Eric Martin	85	5
Satoshi Kobayashi	38	6
Thomas Hermann	47	9
Gang Luo	47	9
David E. Goldberg	231	3
Rakesh Kumar	96	12
Shu Lin	76	2
Richard Taylor	35	16
Cheng Chang	27	5
Lei Jin	20	8

Table 1: Statistics of publications and ambiguous authors

Table 1 presents a list of authors, showing the number of people sharing the same name, and the number of publications attributed to that name.

3. PROBLEM DEFINITION & FEATURES SELECTION

To analyze and accurately assign citation records to their true authors, we first investigate any associative pieces of information that DLs provide along with a citation record. We describe these various pieces of data as "features", and we believe these features, if exploited and used correctly, can be very useful for solving the problem at hand. We try not to restrict our approach to a particular DL; instead we aim to find a set of common features that almost all DLs share. In this section, we present and elaborate on selected common features. Of the selected features, some are directly associated with an author and others with a citation record. We use the term "citation record" to generally represent various types of scientific resources, e.g., conference proceedings, journals papers, patents, books, etc.

Before going further with our description, we first formally define the problem. Suppose that given an author name as the input query, a selected DL returns all citation records that satisfy the query: "author name query $= Q_{authorName} =$ "$Muhammad\ Imran$"".

The results is a set of citation records, $CR = \{cr_1, cr_2, cr_3...cr_n\}$, where cr_i represent a complete record consists of all available features that come with the results. The features set associated with a citation records represented as $CR_i = \{X_1, X_2, ...X_n\}$. The features X_i can be stored in an $N \times D$ matrix, called a feature matrix, where N is the number of entries/citation records (rows) and D is the number of features (columns). For instance, $X_{i,j}$represents jth feature of

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

[2] http://dl.acm.org/

[3] http://scholar.google.com/

[4] http://www.kaggle.com/c/kdd-cup-2013-author-paper-identification-challenge

CR_i citation record.

Given a set of citation records CR, we intend to (1) populate the feature matrix with correct information and (2) produce a list of correct citation records for the author that the user is looking for, while discarding the irrelevant records. To this end, we primarily use common features and their heuristics that are of type generic (i.e., almost all DLs support them, and for some missing features we query the Web). In the next section we describe these features and their heuristics.

4. ACCREDITED FEATURES

4.1 Co-Author

Co-author is defined as a joint author of a book, paper or journal who has collaborated with others to bring about a particular publication. In this paper, we refer to the first author of a citation record as "principal author" and rest of the authors as "co-authors". Thus, a co-author instance represents one author among many co-authors other than the principal author. This is an important, quite useful feature in any DL. The heuristic associated with this feature and employed by our algorithm assumes that if a co-author appears in two different publications with the same principal author, then it is most likely that both publications belong to that principal author. We consider the co-author feature to be a strong ground truth, unless many other features collectively nullify this assumption. It is much less likely that the same co-author collaborates with two different authors (i.e., principal authors) who have the exact same name.

The co-author feature entry in the feature matrix (as described above) represented as:

$$X_i^{type=coAuthor} = \{A_1^p, A_2^c, A_3^c, \dots A_n^c\}$$

where A_i^p represents principal author that also appears as first author of a citation record. And, A_i^c represents a simple co-author of a citation record.

4.2 Conference Venue

Venue represents an event name for a conference or workshop, or a corresponding journal to which researchers submit their publications. These events organize and target one discipline (i.e., computer science, physics, etc.) at a time. And most of the time, an event focuses on a very specific sub-discipline within these broad disciplines. For instance, in the field of computer science a number of events target only the database community, some others target only computer security, and so on. The conspicuous advantage of the venue information lies in the fact that it is an essential part of a citation record and every DL provides it. The heuristic we contrive assumes that a researcher, throughout his career, tends to publish in certain specific sub-disciplines, or related fields under one area of science. It is rarely observed that a researcher publishes in cross disciplines; however many researchers do change their sub-disciplines, but not the general area. Thus, the venue information for two researchers having the same name can be used to differentiate one from the other, based on an examination of the researchers' interest in the identified disciplines and sub-disciplines.

The venue information is represented as a string literal in the feature matrix. For example:

$$X_i^{type=venue} = \{Joint\ Conference\ on\ Digital\ Libraries\}$$

4.3 Authors' Affiliation

Affiliation describes the relationship of an author with an institute, organization, or university to which he/she belongs or works. Providing affiliation information when submitting a publication to a conference, journal or a workshop etc., is obligatory and the data serves as publication metadata. The affiliation of an author tends to change over time. Therefore, the heuristic we use assumes that if two publications with same principal author names also share the same affiliation information, then the publications will be assumed to have the same author.

The affiliation information is represented as a string literal in the feature matrix. For example:

$$X_i^{type=affiliation} = \{QCRI\}$$

4.4 Authors' Names

Name instances are name variations and abbreviations that may be associated with authors, which are based on the size of an author name. For instance, "Muhammad Imran", can be written as "M. Imran", "Imran. M", or "Imran Muhammad". It is evident that many conferences use different name abbreviation sequences for an author, ultimately making it confusing and difficult for DLs to correctly identify and associate publications with their actual authors. The name variation problem is even worse when two authors' shortened name prefix or suffix lead to two completely different names. For instance, "M. Imran" can be "Malik Imran" or "Muhammad Imran" or "Mehmood Imran", which are all true cases. Thus, it is very important to consider this aspect during the disambiguation process. In our approach we keep track of these name variations and resolve ambiguities among them at various stages that are described in Section 5.

The authors' names information is represented as a list of all possible name instances of an author in the feature matrix. For example:

$$X_i^{type=authorName} = \{N_{v=1}, N_{v=2}, N_{v=3}, \dots, N_{v=n}\}$$

where $N_{v=i}$ represents a possible name instance of an author.

4.5 Publication Title

Title is a string literal, i.e., the name of a citation record. Title contains important keywords that can be used for similarity checking between two citation records. A list of extracted keywords of high importance from a set of citation records for an author can be constructive to understand his area of interest, and is quite helpful in mining the similar interest of other authors who share the same name. This feature is used to improve the quality of our results, and that is the reason why we apply this feature at the stage when we confirm an author's set of citation records.

The publication title information is represented as a string literal in the feature matrix. For example:

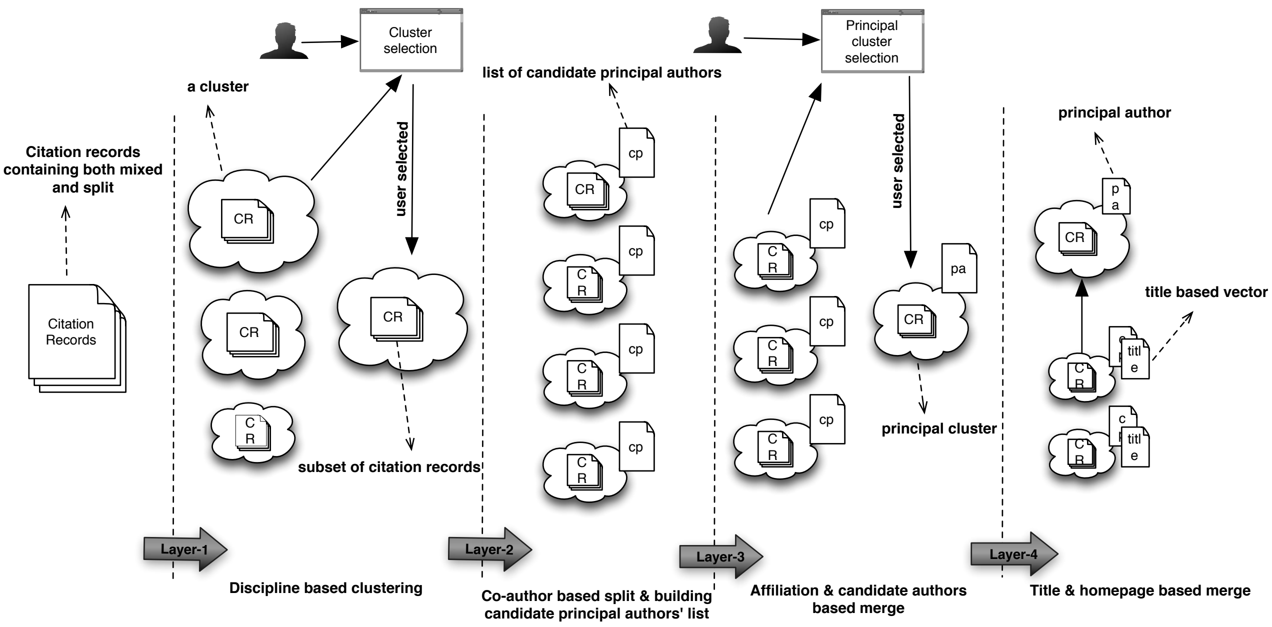
Cluster selection
a cluster
Citation records containing both mixed and split
Citation Records
CR
user selected
subset of citation records
list of candidate principal authors
cp
Principal cluster selection
pa
principal cluster
principal author
title based vector
title
Layer-1
Layer-2
Layer-3
Layer-4
Discipline based clustering
Co-author based split & building candidate principal authors' list
Affiliation & candidate authors based merge
Title & homepage based merge

Figure 1: Name disambiguation: an overall approach

$X_i^{type=title} = \{Correlating\ Independent\ Schema\ Mappings\}$

4.6 Principal Author's Homepage

Homepage is the URL of an author's homepage. If an author has a homepage then this feature can be used as a final step that can increase the quality of the results.

The homepage information is represented as a string literal in the feature matrix. For example:

$X_i^{type=homepage} = \{www.mimran.me\}$

5. DISAMBIGUATION USING MULTILAYER CLUSTERING

5.1 Approach overview

Given the set of selected features (as described in the previous section) we now present the solution to the problem, using unsupervised technique. We use a multilayer hierarchical clustering approach based on divisive approach, the K-means algorithm and agglomerative approach. We form clusters of retrieved citation records containing both mixed and split records, and run them against a user query and disambiguate them at each layer. These layers split the data into meaningful sub-clusters according to the heuristics described in section 4.1, where selected features initially form relatively big clusters (i.e., based on disciplines) and then we divide them into small clusters (i.e., based on co-authors) by employing the K-means algorithm and then join them together using remaining features. Two clusters are merged (i.e., based on agglomerative approach) once the distance between them reduces and reaches a certain threshold.

The similarity measure between the features is determined using Levenshtein Distance algorithm [7]. At certain stages, we also involve the user's feedback on the resultant clusters to get a point to which our algorithm must perform clustering (i.e., combine, split). To this end, using a web interface we show multiple lists of citation records for the user, where each list represents a cluster that is later employed by the user to select his/her desired list. This enables us to limelight the properties of the principal cluster (i.e., a cluster that contains citation records of a principal author) that ultimately leads the disambiguation algorithm to a convergence point. An epitomized view of the overall approach is presented in Figure 1, and in the next section we elaborate on each step in detail.

5.2 Multilayer Hierarchical Clustering

As stated earlier, our algorithm follows a conspicuous multilayer hierarchical clustering approach that divides the whole process into multiple layers in a sensible way that produces quality results. Layers are dependent on each other due to the filtering mechanism and follow a sequential order; a subsequent layer in the sequence uses the results produced by the previous layer, thus improving quality and making each cluster more informative and richer with every step. Furthermore, adaptability of the algorithm allows any other layer, except the first two basic ones (i.e., disciplinary and co-author) to be skipped due to unavailability of meta-data information.

We begin with our definition of a *cluster*, which is:

Cluster: A cluster C consists of those citation records CR whose features' $(X_{i..j})$ similarity measure stays lower than a selected threshold.

$C = \{CR_1, CR_2, \ldots, CR_n\}$, where
$Distance(CR_1^{X_{i..n}}, CR_2^{X_{i..n}}) <= Threshold_{value}$

The following subsections present our algorithm details.

5.2.1 Inter-related disciplines based formation of clusters

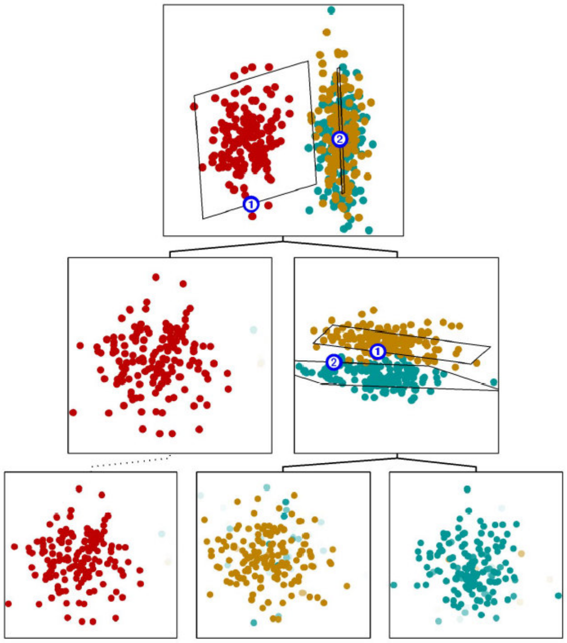

Figure 2: K-Means Clustering: Co-Author's Feature

The first step involves the construction of an initial set of clusters, where we attempt to identify outliers (i.e., dividing records that belong to various different disciplines). This layer is comprised of relatively big clusters, which to a large extent contain all those inter-related citation records that share a common discipline (e.g., computer science, physics, etc.). To form such clusters, we exploit Venue information from the citation records. Emergent clusters after this step must contain all citation records that belong to a common discipline or to a sub-discipline. This means two authors who share a common name but work in two different disciplines should be separated after this step. Finally we present the results (i.e., list of clusters) to the user and ask the user to select his/her desired list. The user intervention at this stage is important because choosing the wrong cluster would never lead the algorithm to converge.

Algorithm 1 shows the steps that we perform during this phase. After the initialization of feature matrix and the assignment of feature value (line 1 & 2), the algorithm checks to see if a cluster with the same venue information as the citation record under consideration (line 5) exists. If there is, it adds the citation record (line 6) to the existing cluster. If not, a new cluster is created with new venue information (line 9, 10 & 11). The resultant list of clusters is then presented to the user to get user's feedback.

5.2.2 *Co-authors based split*

Once the user's feedback is recorded, we proceed to the next step with only one big cluster (i.e., user's selected one). This step involves splitting the cluster into sub-clusters based on the co-authors feature. We employ K-means clustering, which assigns each citation record within the user's selected cluster to one of each K *clusters*, whose center is defined from the co-author's feature space.

Let $C_{user'sselected} = \{CR_1, CR_2..CR_n\}$ be the citation records to be clustered.

```
   Data: Citation Records -> $CR = \{cr_i^n\}$
   Result: List of clusters -> $List < VC >$ venue based clustering
           output
1  $FM = matrix(N \times D)$;      // defining feature matrix to store
   citation records feature values
2  $FM = preprocessing(CR)$;  // performing preprocessing, that is
   to extract related features from the citations records and to
   store into the feature matrix
3  $List_{venue} = array()$;
4  foreach $cr_i \in CR$ do
5      if $getSimilarity(List_{venue}, cr_{i.venue}) >= threshold_{value}$ then
6          $VC_i = cr_i$; // insert citation record in existing cluster
           with same type of venue
7      end
8      else
9          $new\ C = cr_{i.venue}$;                // creating new cluster
10         $C = cr_i$; // adding citation record to newly built cluster
11         $List < VC >= C$;       // adding cluster to venue based
           clusters list
12     end
13 end
14 return $List < VC >$;
```

Algorithm 1: Venue based clustering

And, $\mu = \{\mu_1, \mu_2....\mu_n\}$ be the K clusters centers, which are selected randomly from the list of available feature list(co-author's list), i.e. from the user's selected cluster. For instance,
$K_1 = [coauth1, coauth2, coauth5]$,
$K_2 = [coauth6, coauth7, coauth4, coauth10]$,
$K_3 = [coauth8, coauth10, coauth16]$.

The minimize distortion function describes the distance measure between a citation record and the cluster center μ, and is an indicator of the distance of the n citation records from their respective cluster centers.

$$\jmath(\mu, CR) = \sum_{j=1}^{k} \sum_{i=1}^{N} \left\| CR_i^{(j)} - \mu_j \right\|^2$$

The value of K is evaluated after regress testing and computed according to the number of citation records N_{cr} in user's selected cluster.

$$K = \begin{cases} \frac{2N_{cr}}{5} & : \text{if } N_{cr} \geq 10 \\ 4 & : otherwise \end{cases}$$

This layer involves recursive K-means clustering by using a direct and indirect co-authorship graph. The concept of indirect co-authorship originated from the phenomenon of "six degrees of separation", which proposes that two people from a country or organization are always connected through approximately 6 intermediate acquaintances, implying that we live in a small, interconnected world [6]. This phenomenon also applies in DBLP, and it has been proven that the average distance of authors in a DBLP network has tended to be about 6 for the last 15 years. Moreover, it describes the closeness centrality – how close an author is on average to all others working in similar fields.

The closeness of node v in a connected graph G is as

$$c(v) = \frac{n-1}{\sum_{v,w \epsilon G} d(v,w)}$$

Where v,w belongs to graph G and $d(v, w)$ is the pairwise shortest distance, and n is the number of all nodes reachable from v in G.

This layer also includes maintaining a list of candidate principal authors based on the name variations presented in various citation records. Thus, each resulting cluster is equipped with a list of candidate principal authors that are used in the next steps. Figure 2 illustrates the hierarchical flow of K-means co-author based clustering.

5.2.3 Affiliation based agglomerate

At this point, we have a fair number of clusters, and in order to strengthen the credibility of the contents of each cluster, we employ an affiliation information (i.e., candidate principal authors' affiliation) check across the clusters. We first collect the affiliation information from each sub-cluster for comparative analysis. This task is complicated by the fact that in some cases, researchers use only a university name as an affiliation string, and in others a combination of university and department names, thus making comparison analysis difficult. To address this issue, as stated above, we use *Levenshtein distance* [7] between two different strings to measure the similarity. We then match affiliation information across clusters to merge closely similar ones, which reduces the number of clusters, making them more obvious and unambiguous.

To identify the correct principal author, during the merging process we also check the candidate principal authors list and replace short name variations of an author when a match is found in a cluster that is being compared. At the end of this stage, we only have those clusters that contain citation records having common co-authors, common candidate principal authors, and common affiliation information, a consolidation that improves a cluster's viability. We then again present the remaining clusters to the user, along with detailed information from the citation records, so that the user can select the one among them which most closely matches his choice. After the user's endorsement, the selected cluster from this stage will be considered as the principal cluster, having the principal author and containing accurate information.

5.2.4 Pursuit of the remaining bits

Although the previous layers deliver quite impressive results, in cases where information is missing, and to further ensure the creditability of the resulting clusters, this layer strives to identify any remaining citation records that may be split among other clusters. To achieve this, we use the title and homepage features. To check the title similarity, we build a vector representing keywords and their frequencies in all the citation records in a cluster. Prior to the vector building we perform pre-processing steps (e.g., stop-words filtering, lemmatization, etc.), because authors use various forms of the same word in publication titles, such as 'organize', 'organizes', and 'organizing', which makes it hard to get a good comparison score. To address this we rely on lemmatization, which helps to match a word in one morphological form in a title to another morphological form. We perform these steps for the principal as well as for all other clusters.

A similarity measure process runs between the vector of the principal cluster and the vectors of all other clusters, one by one. If a cluster is found with greater similarity, then the citation records will be merged in the principal cluster. As a final step, we check the homepage of the principal and the other candidate principal clusters authors. We use the Google search API for this purpose. If within the first ten organic search results we find the principal author name having as a keyword "homepage", and most of the keywords present in the vectors built during the previous step, then that is considered to be the homepage of the author, and its URL is maintained. The same process is performed for all the other clusters' authors and we merge those citation records with a homepage URL that is the same as the principal author's URL. We note that this feature is very useful; however, we do not base our entire algorithm on this feature because many new researchers do not have a homepage.

6. EXPERIMENT & EVALUATION

This section details the experimental procedure that was conducted using a set of researchers with ambiguous names to evaluate our proposed method. All the experiments were performed on the DBLP data set, which contains more than 2 million publications. We chose ADANA [20] as our baseline approach because their method offers the most convincing results compared to the others, as stated in the related work section. In order to evaluate our algorithm with this baseline approach, we collected 50 of the most ambiguous names that have also been used in [20, 1]. We manually annotated the golden dataset by identifying their homepages and maintaining a correct citation records list for each researcher locally. For each researcher in the golden dataset we queried DBLP and obtained the citations records. These citation records contain both mixed and split records. Then, each researcher's citation records were given to our disambiguation algorithm, which eventually returned a list of disambiguated records.

The results obtained from the experiments were evaluated according to traditional measures such as precision, recall and the F1 score as follows:

$$Precision = \frac{tp}{tp+fp} \quad Recall = \frac{tp}{tp+fn} \quad F1 = \frac{2 \times Percision \times Recall}{Percision + Recall}$$

Where t_p is true positive, i.e citations records are written by the same author and the result return by the algorithm is right, f_p false positive: citation records are written by different people while algorithm returns as belongs to same author, and false negative f_n: citation records are written by the same author while algorithm resolve it as they are written by different author.

6.1 Experiment Results

Table 2 shows the results of our experiment on the golden dataset showing Recall, Precision and F1. From our experiments we concluded that on average, for various researcher names, 70% of the disambiguation was resolved with the co-authorship phase, around 20% by affiliation matching and 10% using title homepage matching.

Table 3 shows the geometric means of precision, recall and F1 values of both ADANA and our algorithm. The computed average results for our method demonstrates a 8.024%

Author	*Rec.*	*Prec.*	*F1*	*Author*	*Rec.*	*Prec.*	*F1*
Lu Liu	1.00	1.00	1.00	Hui Fang	1.00	1.00	1.00
Qiang shen	1.00	0.97	0.98	Peter Phillips	0.90	1.00	0.94
Charles Smith	1.00	1.00	1.00	Thomas Meyer	0.85	1.00	0.91
Xiaoming Wang	1.00	1.00	1.00	Paul Brown	1.00	1.00	1.00
Satoshi Kobayashi	1.00	0.73	0.84	Steve King	0.55	1.00	0.71
Thomas Hermann	1.00	1.00	1.00	Yun Wang	1.00	0.94	0.97
Cheng Chang	1.00	0.85	0.92	David E. Goldberg	0.97	1.00	0.99
Rakesh Kumar	0.81	1.00	0.90	Jim Gray	0.80	1.00	0.89
David Levine	0.82	1.00	0.90	Bin Zhu	0.90	0.64	0.75
Bob Johnson	1.00	1.00	1.00	Pillip j. Smith	0.92	1.00	0.96
Jing Zhang	1.00	0.92	0.96	Fan Wang	0.85	1.00	0.91
Paul Wang	1.00	1.00	1.00	Kai Tang	1.00	0.97	0.99
Wei Xu	0.97	1.00	0.98	Thomas Zimmermann	1.00	0.97	0.98
William H. Hsu	0.92	1.00	0.95	Eric Martin	1.00	1.00	1.00
Lei Jin	1.00	1.00	1.00	Li Shen	0.90	1.00	0.94
Lei Chen	0.95	1.00	0.97	J. Guo	1.00	1.00	1.00
Ji Zhang	1.00	0.92	0.95	Gang Luo	1.00	0.96	0.98
R. Ramesh	1.00	1.00	1.00	Feng Pan	0.76	1.00	0.86
Thomas D. Taylor	1.00	1.00	1.00	Juan Carlos Lopez	1.00	0.97	0.98
Young Park	0.85	1.00	0.92	Michael Wagner	0.97	1.00	0.98
Yong Chen	0.96	0.87	0.91	Yoshio Tanaka	1.00	0.93	0.96
Z. Wang	0.88	1.00	0.94	F. Wang	1.00	1.00	1.00
Alok Gupta	0.93	1.00	0.96	Ping Zhou	1.00	1.00	1.00
Michael Lang	0.88	1.00	0.93	Frank Mueller	0.91	1.00	0.95
Mark Davis	0.85	1.00	0.92	Xiaoyan Li	1.00	1.00	1.00

Table 2: Results for the 50 Authors Names on Golden Data Set.

Measures	*ADANA*	*Our Algorithm*
Precision	0.95	0.95
Recall	0.837	0.94
F1	0.89	0.95

Table 3: Geometric Mean's Comparison

increase in the F1, a 13.23% increase in recall, and quite similar precision values.

7. RELATED WORK

Over the last few years, many approaches have been proposed for solving the name disambiguation problem. The early adopted methods included authorship attribution [11], resolution and word sense Disambiguation [14] based on the Stylometry measures of the text, then record Linkage [5] where the names, addresses, phone numbers, genders and dates of birth of individuals were used to link their records. Bibliometric fingerprints [16] tackled name disambiguation problems related to the common surnames in China. The algorithm adapted is based on approximate structural equivalence (ASE), where articles sharing a certain number of references, or a rare reference, with the same or similar author name, are treated as being written by the same author. In this technique, the importance of a reference is also considered, which is calculated by using a knowledge homogeneity score (KHS). Because it operates as a single layer architecture it is prone to errors when the KHS parameter is not set right, thus resulting in two authors with similar names and similar fields of interest being treated as one person.

The later techniques consist of classification of systems into supervised and unsupervised learning. These techniques [15, 9, 8, 19] are based on supervised classification and based on both a generative (naive Bayes) and discriminative model (Support Vector Machines). They divide the work into classes of features that are based on intuition and take into account (i) the relationship between name variations and citations, and (ii) topic consistency for an author, but these techniques require a big dataset to train the algorithm.

The techniques described in [20, 10, 18, 17] are based on unsupervised learning and use feature values to disambiguate publications. [17] perform author name disambiguation considering not only known links between a pair of articles, but some implicit relationships as well. It employs probability analysis to compute the maximum likelihood of merging clusters together, but fails to cater to different variants of names and is known to have some computational issues. [18] is directed mainly towards homonymy and classified as semi-supervised because it manually sets a parameter called low redundancy cut-off. This parameter is based on the fact that most of the author names with the same last name and different initials represent the same individual, thus avoiding having the algorithm affected by the disambiguation process in a negative manner. The drawback of this technique lies in the low redundancy cut-off parameter allowing different authors with the same last name to be treated as one person. ADANA [20] is the closest method to our technique and also implements users' feedback in the resulted output. The method uses the pairwise factor graph model (PFG) to model the observable variables (pairs of articles) for the best fit, such that the variables whose values, based on the feature set relationship, decide whether the pair of articles belong to the same cluster or not, which maximizes the objective function although it is restricted to a specific data model and not adaptive in nature.

These techniques are primarily of a specific type that target a specific problem within this domain, and fail to provide a generic solution that could also consider the name variations aspect and users' feedback to help the system to completely disambiguate researchers' names.

8. CONCLUSION AND FUTURE WORK

The ever increasing number of researchers and bibliographic citations poses serious challenges for digital libraries that tackle the problem of name disambiguation. Lack of standards for using researchers' names, misspellings, different name variations, and sharing common names, create ambiguities and hinder the correct identification of citation records for a given researcher. In this paper we presented an end-to-end system that performs retrieval of data from digital libraries and disambiguates them against a user query. In this approach we exploited features from both author and citation records. Moreover, the unsupervised approach, with the help of user interventions at certain stages, contributed substantially to achieving high quality results. The experiments on a set of researchers' names that were considered to be highly ambiguous decisively produced high precision and recall results, and affirmed the viability of our algorithm.

Our system works as a wrapper service on top of DLs. The disambiguation service can also be used by a third-party system. For instance, we aim to use this service as a service component in our ResEval Mash platform [13]. The ResEval Mash platform is geared towards non-programmers to help them accomplish complex research evaluation tasks.

9. REFERENCES

[1] B. C. . O. E. Arms, W.Y. An architecture for information in digital libraries. *D-Lib Magazine*, 3, 1997.

[2] J. Artiles, J. Gonzalo, and 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*, SemEval '07, pages 64–69, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[3] M. Atay, Y. Sun, D. Liu, S. Lu, and F. Fotouhi. Mapping xml data to relational data: A dom-based approach. *CoRR*, abs/1010.1746, 2010.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] C. Day. Record linkage i: Evaluation of commercially available record linkage software for use in nass. *STB Research Report*, STB-95-02:1–12, 1995.

[6] E. Elmacioglu and D. Lee. On six degrees of separation in dblp-db and more. *ACM SIGMOD Record*, 34(2):33–40, 2005.

[7] R. Haldar and D. Mukhopadhyay. Levenshtein distance technique in dictionary lookup methods: An improved approach. *CoRR*, abs/1101.1232, 2011.

[8] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *In JCDL '04: Proceedings of the 4th ACM/IEEE joint conference on Digital libraries*, pages 296–305, 2004.

[9] H. Han, W. Xu, H. Zha, and 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*, SAC '05, pages 1065–1069, New York, NY, USA, 2005. ACM.

[10] H. Han, H. Zha, and 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*, JCDL '05, pages 334–343, New York, NY, USA, 2005. ACM.

[11] P. Juola. Authorship attribution. found.trends inf. *Retr*, 1:233–234, 2006.

[12] D. Lee, B. won On, J. Kang, and S. Park. Effective and scalable solutions for mixed and split citation problems in digital libraries. In *INFORMATION QUALITY IN INFORMATIONAL SYSTEMS*, pages 69–76. ACM, 2005.

[13] I. Muhammad, K. Felix, S. Stefano, D. Florian, C. Fabio, and M. Maurizio. Reseval mash: A mashup tool for advanced research evaluation. In *WWW2012*, pages 361–364. ACM, 2012.

[14] M. Stevenson and Y. Wilks. Word sense disambiguation. *The Oxford Handbook of Comp. Linguistics*, pages 249–265, 2003.

[15] X. Sun, J. Kaur, L. Possamai, and F. Menczer. Detecting ambiguous author names in crowdsourced scholarly data. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 568–571, 2011.

[16] L. Tang and J. Walsh. Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps. *Scientometrics*, 84(3):763–784, 2010.

[17] V. I. Torvik and N. R. Smalheiser. Author name disambiguation in medline. *ACM Trans. Knowl. Discov. Data*, 3(3):11:1–11:29, July 2009.

[18] T. A. Velden, A.-u. Haque, and C. Lagoze. Resolving author name homonymy to improve resolution of structures in co-author networks. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, JCDL '11, pages 241–250, New York, NY, USA, 2011. ACM.

[19] F. Wang, J. Li, J. Tang, J. Zhang, and K. Wang. Name disambiguation using atomic clusters. In *Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management*, WAIM '08, pages 357–364, Washington, DC, USA, 2008. IEEE Computer Society.

[20] X. Wang, J. Tang, H. Cheng, and P. S. Yu. Adana: Active name disambiguation. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 794–803, Washington, DC, USA, 2011. IEEE Computer Society.