Chapter 4

Metadata-based Author Name Disambiguation & Profile Integration

4.1 Introduction

Distinguishing similar entities is contingent upon the quantum of information available about them: the more the information the easier it is to distinguish them and vice versa [Shin et al. 2014]. Thus attributes or metadata play an important role in distinguishing various entities and grouping or separating them. In this chapter, we discuss a name disambiguation algorithm that uses multiple publication attributes (metadata), including author names, publication venues, titles, etc. for solving the problem. Generally any algorithm dealing with name ambiguity problem performs two major operations: determining similarity between two entities; and, clustering entities on the basis of their similarity. We propose a hybrid clustering mechanism which uses hard clustering in the first stage and then uses soft clustering to cluster citation records.

This Chapter deals with two main issues: resolving the name ambiguity problem in digital citations and profile integration. In Section 4.2 we define name disambiguation and important terms that we use in the rest of this Chapter. Section 4.3 throws some light on clustering methods and their use in name disambiguation. In Section 4.4, we provide an overview of the proposed name disambiguation technique while Section 4.5 presents the experimental results and discussions. In Section 4.6, we present the proposed
profile integration technique and associated results. Section 4.7 concludes the Chapter.

4.2 Preliminaries & Problem Definition

4.2.1 Preliminaries

- **Citation-Record:** A record consisting of various attributes of a publication. The attributes of interest here are the principal author and co-authors, their affiliations, their e-mail IDs, publication title, venue and year of publication. We use Citation-Records and publications interchangeably in this work.

- **Target-Author:** A particular individual (T). A list of Citation-Records can have one or more variations of the name of the Target Author.

- **Atomic-Cluster:** A list of citation records in which intra-cluster similarity is much higher that inter-cluster similarity.

- **Publication-Profile:** A list of citation records containing all the publications of a Target Author.

4.2.2 Problem Definition

Given a set of n citation-records, the goal of the disambiguation process is to create m publication-profiles, with \( m \leq n \), such that \( m_i \cap m_j = \emptyset \) \( \forall i, j \): i.e. none of the \( m_i \) contains any citation-record of any \( m_j \) and each \( m_i \) contains only the publications exclusive to it. The task may seem trivial but the existence of multiple variations of the key attributes of a particular target author and similarity between the key attributes of different target authors in citation-records makes the job very difficult.

4.3 Clustering Methods for Name Disambiguation

Clustering is an ancient technique and derives inspiration from human need for identifying men and objects based on their salient characteristics [Arif et al., 2014b]. Though clustering techniques have been widely used in
modern day computing applications like pattern recognition [Anderberg, 1973], image processing [Flynn et al., 1991] and information retrieval [Rasmussen, 1992; Salton, 1991], clustering has been able to make a dent in other disciplines also [Jain and Dubes, 1988].

There are numerous situations where there is very little information available about the structure of data on the basis of which it could be analysed. In such situations clustering becomes an efficient tool as it does not rely on the assumptions underlying common statistical methods. It is therefore useful in applications for pattern discovery and analysis, grouping of objects, decision making, data mining, document retrieval, and image segmentation [Jain et al., 1999].

The clusters generated by a clustering technique can be disjoint or overlapping depending upon whether the technique used is crisp (hard) or fuzzy. In hard clustering the membership of a data item to a cluster is defined by the set {0, 1}, i.e. it either belongs to a cluster or does not. In fuzzy clustering the data item belongs to several clusters simultaneously with certain degree of membership in the range [0, 1] and the sum of all the memberships of a data item to these clusters is equal to 1. Hard and fuzzy clustering are based on the classical and fuzzy set theories, respectively [Zadeh, 1965]. The advantage provided by the fuzzy clustering techniques can be put to use efficiently by designing efficient membership functions [Rokach and Maimon, 2005].

Given a data set Z with n elements, hard clustering partitions Z into C clusters or partitions, where C ≤ n. The number C may be specified externally or may be generated during run time. In terms of classical set theory, the clusters generated in hard clustering satisfy the following conditions [Bezdek, 1981]:

$$\bigcup_{i=1}^{C} P_i = Z,$$

(4.1)
Chapter 4: Metadata-based Author Name Disambiguation & Profile Integration

\[ P_i \cap P_j = \emptyset, \quad 1 \leq i \neq j \leq C, \]  
\[ \emptyset \subset P_i \subset Z, \quad 1 \leq i \leq C \]  

Analogously these conditions can also be expressed in terms of membership functions \( \mu \). Each row of the partition matrix \( U = [\mu_{ik}]_{C \times n} \) corresponds to the values of membership function \( \mu_i \) of the \( i \)th subset \( P_i \) of \( Z \). In this case the conditions specified in Equations 4.1, 4.2, and 4.3 can be expressed in terms of Equation 4.4, 4.5, and 4.6 respectively.

\[ \mu_{ik} \in \{0,1\}, \quad 1 \leq i \leq C, \quad 1 \leq k \leq n, \]  
\[ \sum_{i=1}^{C} \mu_{ik} = 1, \quad 1 \leq k \leq n, \]  
\[ 0 < \sum_{k=1}^{n} \mu_{ik} < n, \quad 1 \leq i \leq C \]  

The hard partitioning space \( M_{hc} \) of all the possible partition matrices of \( Z \) is expressed in (4.7) [Bezdek, 1981]:

\[ M_{hc} = \left\{ U \in \mathbb{R}^{C \times n} | \mu_{ik} \in \{0,1\}, \forall i,k; \sum_{k=1}^{C} \mu_{ik} = 1, \forall k; 0 < \sum_{i=1}^{n} \mu_{ik} < n, \forall i \right\} \]  

On the other hand, if the condition \( \mu_{ik} \in \{0,1\} \) in Equation 4.4 is replaced by \( \mu_{ik} \in [0,1] \), the hard partitions get converted to fuzzy partitions. Thus, the partition matrix expressed by Equations 4.4, 4.5, and 4.6 as hard can be expressed [Ruspini, 1969] in terms of Equations 4.8, 4.9 and 4.10, as fuzzy.

\[ \mu_{ik} \in [0,1], \quad 1 \leq i \leq C, \quad 1 \leq k \leq n \]  
\[ \sum_{i=1}^{C} \mu_{ik} = 1, \quad 1 \leq k \leq n \]
The condition in Equation (4.9) specifies that the sum of membership values of a data item to various clusters (c₁, c₂, ..., cₙ) is equal to 1. Analogously, the fuzzy partitioning space \( M_{fc} \) of all the possible partition matrices of \( Z \) is expressed as:

\[
M_{fc} = \left\{ u \in \mathbb{R}^{C \times n} | \mu_{ik} \in [0, 1], \forall i, k; \sum_{k=1}^{C} \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^{n} \mu_{ik} < n, \forall i \right\} \quad (4.11)
\]

Commonly used clustering techniques in name disambiguation task are: partitioning [Han and Kamber, 2005], which creates a pre-defined number of partitions \( K \) of the set of publication-records; hierarchical agglomerative clustering [Han and Kamber, 2005], which groups the publication-records to authors in a hierarchical manner; density-based clustering [Han and Kamber, 2005], in which a dense region corresponds to a cluster of publication-records of an author; and spectral clustering [Zha et al., 2001], which uses graph-based clustering techniques.

Agglomerative clustering (hierarchical) has been used by [Bhattacharya and Getoor, 2007; Cota et al., 2010; Culotta et al., 2007; Kang et al., 2009; Pereira et al., 2009; Soler, 2007; Song et al., 2007; Torvik and Smallheiser, 2009; Liu et al., 2014a; Liu et al., 2014b]; partitioning clustering by [Kanani et al., 2007; Yang et al., 2008]; density-based clustering by [Huang et al., 2006]; and, spectral or graph-based clustering by [Han et al., 2005; Fang et al., 2011; Wang et al., 2012; Shin et al. 2014]. Soler [2007] uses a mix of both spectral and agglomerative clustering.

Some of the existing name disambiguation techniques use two stage clustering mechanism for disambiguating authors. One such method [Obara and Miyamoto, 2012] uses a combination of two hard clustering techniques i.e. partitioning and hierarchical, where K-means [MacQueen, 1967] is used in the first stage and agglomerative hierarchical clustering in the next. Hard
and soft clustering methods have been used by [Ikeda et al., 2009] to disambiguate person names on the Web in a two stage clustering framework. Ikeda et al. [Ikeda et al., 2009] use hard hierarchical clustering in the first stage and soft hierarchical clustering in the second stage.

4.3.1 Hierarchical Clustering

Hierarchical clustering is of two types: divisive and agglomerative. In divisive approach the clustering algorithm starts with all the data items in a single cluster. Based on the similarity or distance, the initial cluster is divided into sub-clusters iteratively until a desired clustering structure is obtained. However, the reverse happens in agglomerative clustering, each data item begins as a separate cluster or more specifically, a singleton cluster. These clusters are then merged iteratively, based on the similarity or distance measure, until a desired clustering structure is obtained [Jain et al., 1999].

Of the hierarchical-clustering based author name disambiguation methods proposed so far, none has used divisive hierarchical clustering [Ferrira et al., 2012]. However agglomerative hierarchical clustering has been used widely [Ferrira et al., 2012] as also evidenced from a number of agglomerative clustering based name disambiguation methods cited above.

4.3.1.1 Hierarchical Agglomerative Clustering (HAC)

Hierarchical Agglomerative Clustering (HAC), first proposed by S. C. Johnson in 1967 [Johnson, 1967], has been widely used in a variety of applications. Agglomerative clustering uses a bottom-up approach for creation of clusters in contrast to divisive clustering, which uses a top-down approach. A hierarchical agglomerative clustering algorithm for a given dataset $D$ is presented in Figure 4.1.
Algorithm: Hierarchical-Agglomerative-Clustering

1. For all initial singleton clusters compute the pairwise similarity based on a similarity/distance measure.
2. Create bigger clusters by merging clusters with distance below a particular threshold.
3. Compute pairwise similarity between clusters obtained in Step 2.
4. Repeat Steps 2 and 3 until all the data items are grouped in one cluster or some other termination criteria is fulfilled.

Figure 4.1: Hierarchical Agglomerative Clustering algorithm.

For any two data points \( a \) and \( b \) in a dataset \( D \), the similarity value in Step 1 and Step 3 of the above algorithm can be calculated by using any of the following distance/similarity measures:\(^{19}\):

(i) \( \text{Euclidean Distance: } \|a - b\|_2 = \sqrt{\sum (a_i - b_i)^2} \) (4.12)

(ii) \( \text{Manhattan Distance: } \|a - b\|_1 = \sum |a_i - b_i| \) (4.13)

(iii) \( \text{Mahalanobis Distance: } \sqrt{(a - b)^T S^{-1}(a - b)} \),
where \( S \) is Co-variance Matrix

(iv) \( \text{Cosine Similarity: } \frac{a \cdot b}{||a|| ||b||} \) (4.15)

The similarity between any two clusters \( c_i \) and \( c_j \) in hierarchical agglomerative clustering can be calculated in three different ways viz. single-link, complete-link and average-link [Jain et al., 1999]. If the shortest distance between a data item in \( c_i \) and a data item in \( c_j \) is considered as the distance between \( c_i \) and \( c_j \), it is single-link clustering and is calculated using the

formula in Equation 4.16. On the other hand, if the longest distance between a data item in \( c_i \) and a data item in \( c_j \) is considered as the distance between \( c_i \) and \( c_j \), it is complete-link clustering and is calculated using the formula in Equation 4.17. However, if average distance between the data items in \( c_i \) and the data items in \( c_j \) is considered as the distance between \( c_i \) and \( c_j \), it is average-link clustering and is calculated using the formula in Equation 4.18.

Single-Link: \[ \min\{d(a, b): a \in c_i, b \in c_j\} \] (4.16)

Complete-Link: \[ \max\{d(a, b): a \in c_i, b \in c_j\} \] (4.17)

Average-Link: \[ \frac{1}{|c_i||c_j|} \sum_{a \in c_i} \sum_{b \in c_j} d(a, b) \] (4.18)

The results of a hierarchical clustering algorithm may be presented in the form of a dendogram.

### 4.3.2 Partitioning Clustering

Sometimes the size of dataset is very large. In such cases hierarchical clustering may not be an efficient choice [Jain et al., 1999]. Partitional clustering methods can be more useful in such cases. Starting with an initial random partitioning, these methods relocate data items between clusters to optimize a criterion function. Partioning methods normally require the desired number of output clusters as an input from the user. This requirement has the potential to affect the entire clustering process. There is evidence in literature of some efforts [Dubes, 1987] to address this issue. Since it is not computationally feasible to enumerate all the possible partitions, certain greedy heuristics in the form of iterative optimizations have been used [Rokach and Maimon, 2005]. There are two categories of partitional clustering algorithms: Error Minimization and Graph-Theoretic.
4.3.2.1 Error Minimization Algorithms

K-means algorithm is the most widely used partitional clustering algorithm which employs a squared error criterion [MacQueen, 1967]. The standard hard k-means algorithm is presented in Figure 4.2.

Algorithm: Hard K-Means

Input: Instance set $S$, number of clusters $K$

Output: Clusters ($K$)

1. Initialize $K$ cluster centers ($m_1, m_2, ...$)
2. while termination condition is not satisfied do
3. Assign instances to the closest cluster center:
   \[ S_i^{(n)} = \{x_j : \|x_j - m_i^{(n)}\|^2 \leq \|x_j - m_j^{(n)}\|^2 \forall j, 1 \leq j \leq k \} \]
   each $x_j$ above is assigned to exactly one $S_i^{(n)}$.
4. Update cluster centers based on the above assignment.
   \[ m_i^{(n+1)} = \frac{1}{|S_i^{(n)}|} \sum_{x_j \in S_i^{(n)}} x_j \]
5. [End while]

Figure 4.2: K-means clustering algorithm.

The fuzzy counterpart of the algorithm in Figure 4.2 i.e. c-means is presented in Figure 4.3.

4.3.2.2 Graph-Theoretic Clustering Algorithms (Spectral Clustering)

In graph-theoretic methods, clusters are generated using the concept of graphs. The instances are represented as nodes in this clustering scheme. While other methods falter in case of arbitrary shaped clusters, spectral clustering methods try to leverage their geometry. Spectral clustering makes
use of affinity matrix instead of working with data directly. The affinity matrix takes into account the non-linear nature of similarity between data.

**Algorithm:** Fuzzy c-means.

**Input:** Instance set \( S \), number of clusters \( c \), weighting exponent \( m \), termination tolerance \( \varepsilon \) and the norm inducing matrix \( A \).

**Output:** Clusters (c)

1. Randomly initialize the partition matrix.
2. while termination condition \( \| U^{(n)} - U^{(n-1)} \| < \varepsilon \) is not satisfied, do
3. Find cluster centres (means) for \( n=1, 2, 3, \ldots \)
   \[
   v_i^{(n)} = \frac{\sum_{k=1}^{N} (\mu_{ik}^{(n-1)})^m s_k}{\sum_{k=1}^{N} (\mu_{ik}^{(n-1)})^m}, 1 \leq i \leq c.
   \]
4. Find distance:
   \[
   D_{ikA}^2 = (s_k - v_i^{(n)})^T A (s_k - v_i^{(n)}), 1 \leq i \leq c, 1 \leq k \leq N.
   \]
5. Update the partition matrix:
   for \( 1 \leq k \leq N \)
   \[
   \mu_{ik}^{(n)} = \frac{1}{\sum_{j=1}^{c} (D_{ijA}/D_{ikA})^{2/(m-1)}} \quad \text{if} \quad D_{ikA} > 0 \forall i = 1, 2, \ldots, c
   \]
   \[
   \mu_{ik}^{(n)} = 0 \quad \text{if} \quad D_{ikA} > 0, \quad \text{and} \quad \mu_{ik}^{(n)} \in [0, 1] \quad \text{with}
   \]
   \[
   \sum_{i=1}^{c} \mu_{ik}^{(n)} = 1.
   \]
   [End for]
   [End while]

Figure-4.3: Fuzzy c-means algorithm.

Figures 4.4 (a) and 4.4 (b) depicts the difference between the clusters of spirals dataset generated by \( k \)-means and spectral clustering respectively. This example has been taken from [Karatzoglou, et al., 2004].
Graph-theoretic clustering methods can be partitioning or hierarchical [Jain et al., 1999]. Zhan et al. [Zahn, 1971] proposed a graph theoretic partitioning clustering algorithm which generates or infers clusters by constructing minimal spanning tree (MST) of the instances. Clusters are generated by deleting the MST with longest edges.

**4.3.3 Density-based Clustering**

Density based clustering methods are considered good at finding clusters of arbitrary shape and are easy to understand [Parimala, et al., 2011]. Of all the density-based clustering techniques proposed so far DBSCAN is the most popular owing to its simplicity and ease of use. It requires only two input parameters: ‘\( \epsilon \)’ which specifies the radius i.e. neighbourhood of expected clusters and ‘MinPts’ which specifies the minimum number of members required in each of the resultant clusters. Unlike the k-means algorithm where the number of resultant clusters has to be provided as an input and the clusters are of spherical shape, DBSCAN does not need this input and can find clusters of any arbitrary size and shape.
4.4 A Novel Name Disambiguation Approach

4.4.1 Attributes of Importance

Table 4.1 presents a list of various components of a citation-record that are important in the name disambiguation technique presented in this work. A brief description of these attributes is given below.

Table 4.1: Attributes of Importance

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_i$.title</td>
<td>Title of the publication</td>
</tr>
<tr>
<td>$c_i$.authors</td>
<td>Set of authors ${a_1, a_2, \ldots}$ of the publication</td>
</tr>
<tr>
<td>$c_i$.e-mail IDs</td>
<td>Set of e-mail IDs ${e_1, e_2, \ldots}$ of authors</td>
</tr>
<tr>
<td>$c_i$.affiliations</td>
<td>Set of affiliations ${x_1, x_2, \ldots}$ of authors</td>
</tr>
<tr>
<td>$c_i$.venue</td>
<td>Conference or Journal of the publication</td>
</tr>
</tbody>
</table>

In total, we consider five attributes. We call a set of these attributes as citation-record. Information about these can be easily obtained from the publications themselves.

4.4.2 Multiple Clustering based Name Disambiguation

This section provides an explanation of the disambiguation approach proposed in this work. Our approach is quite similar to the one followed in [Wang et al., 2008] and [Imran et al., 2013] in terms of creation of an initial set of atomic-clusters. While [Wang et al., 2008] use year of publication for creation of initial set of atomic clusters, [Imran et al., 2013] use publication venue for this purpose. Our technique differs from both these in its use of e-mail ID instead of year-of-publication or publication venue to create the initial set of atomic-clusters. Besides, while both [Wang et al., 2008] and [Imran et al., 2013] use $k$-means for further clustering, we use hierarchical agglomerative clustering to merge clusters using vector space model. Our method is an improvement over [Wang et al., 2008] in that e-mail ID is a better identification measure than year-of-publication and with e-mail ID we
get more precise initial set of clusters. Similarly, this approach returns better initial clusters compared to those obtained by [Imran et al., 2013] using publication venue. Though [Imran et al., 2013] claims to be an unsupervised method, it takes user’s input after the initial set of venue based clusters is generated. Our approach works in an entirely unsupervised manner without any user interference or input during the process.

The proposed approach uses a mix of both hard and fuzzy clustering for author name disambiguation. The use of fuzzy clustering in the second half is primarily to deal with the problem of split citations.

4.4.2.1 Two-Stage Multiple Clustering (TSMC)

Comparing citation-records and clustering them hierarchically on the basis of similarity/distance between them has been the methodology followed in various important studies [Mann & Yarowsky, 2003; Culotta et al., 2007; Song et al., 2007; Torvik & Smalheiser, 2009; Kang et al., 2009; Wang et al., 2012; Liu et al., 2014a; Liu et al., 2014b]. Majority of the approaches to name disambiguation combine the similarity score of all the attributes considered into a single numeric value to decide whether a pair of citation-records belong to the same author or not [Smalheiser and Torvik, 2009].

The proposed model considers the similarity value of each attribute separately in a sequential manner to cluster publications. It is a two stage clustering process. Hierarchical agglomerative clustering [Jain et al., 1999] is used in first stage and fuzzy clustering in the second.

First stage comprises of four clustering steps. Each step produces an intermediate set of clusters on the basis of a particular publication attribute. The first step generates clusters on the basis of e-mail, the second on the basis of affiliation of the target author, the third on the basis of co-authors and the fourth on the basis of publication venue. In order to avoid the problem of transitivity we deliberately use publication venue details (publisher, journal, conference, etc.) in the last step.
Algorithm: Two-stage Multiple Clustering (TSMC)

Input: An Author Name (Target Author \(T\))

Output: A list of Publications-Profile (Atomic Clusters (C) of Citation-Records where each cluster contains publications of an actual author), \(R_f\)

1. Retrieve Citation-Records (CR) from database with \(T\) as one of the authors
2. Extract all e-mail-IDs \(E\) from CR
3. \(R_i\leftarrow\) e-mailID-based clusters (CR, \(E\))
4. \(R_i\leftarrow\) (\(R_i\), Affiliation Information) [affiliation-based clusters]
5. \(R_i\leftarrow\) (\(R_i\), Co-author Information) [co-author-based clusters]
6. \(R_i\leftarrow\) (\(R_i\), Venue Information) [venue-based clusters]
7. \(R_f\leftarrow\) (\(R_i\), Fuzzy Membership Values) [Final clusters]

Figure 4.5: Name Disambiguation Algorithm

After retrieving the publications for a target author name, the proposed algorithm as presented in Figure 4.5, creates a set of publication-profiles i.e. clusters. Each cluster is supposed to contain publications of a separate author. These clusters represent the publication-profiles. In this algorithm \(R_i\) and \(R_f\) represent intermediate and final results, respectively. Step 7, which comprises the Second stage, creates the final set of clusters.

The architecture of the proposed author name disambiguation system is shown in Figure 4.6.

In each of these stages we need to calculate the similarity/distance between any two strings: be they e-mail IDs, affiliations, co-authors or publication venues. In the past, both supervised and unsupervised techniques have been used for calculating this similarity/distance between strings [On et al., 2005]. Detailed discussion on string similarity metrics is given in Section 3.3 of Chapter-3. Unsupervised methods either use string-based distance or vector-based Cosine distance. Jaccard-Coefficient, TF-IDF, Jaro and Jaro-Winkler are string-based distance methods. However, in case of vector-based Cosine distance the angle \((\theta)\) between two vectors is
calculated. Since we use Jaro-Winkler and Cosine-similarity for comparing two citation-records, we define them as in Equation 4.20 and 4.21, respectively. Equation 4.19 is used to define Equation 4.20. In each of these equations \( cr \) and \( c \) represent a citation-record and a cluster, respectively.

\[
J_{\text{aro}}(cr, c) = \frac{1}{3} \left( \frac{|C_{c} \cap C_{r}|}{|C_{r}|} + \frac{|C_{c} \cap C_{r}|}{|C_{c}|} + \frac{|C_{c} \cap C_{r}| - T_{c,c,r,c}}{2|C_{c} \cap C_{r}|} \right)
\]

(4.19)

\( C_i \) “all characters in \( i \)”,
\( CC_{jk} \) “all characters in \( j \) that appear in \( k \)”, and
\( T_{lm} \) “number of transpositions of characters in \( l \) relative to \( m \)”

Figure 4.6: Architecture of the proposed Multiple Clustering based Name Disambiguation system
\[ \text{Jaro-Winkler}(cr, c) = \text{Jaro}(cr, c) + \frac{\max(L, 4)}{10} \times (1 - \text{Jaro}(cr, c)) \]  

(4.20)

where \( L \) is the length of common prefix of \( cr \) and \( c \).

\[
\text{CosineSimilarity}(cr, c) = \cos(\theta) = \frac{cr \cdot c}{||cr|| ||c||} = \frac{\sum_{i=1}^{n} c_i \times c_i}{\sqrt{\sum_{i=1}^{n} (cr_i)^2} \times \sqrt{\sum_{i=1}^{n} (c_i)^2}}
\]  

(4.21)

The proposed method is different from existing methods in that instead of downloading and using entire dump of DBLP dataset, it obtains only the citation records with a given author name from the DBLP database. The records returned are parsed and metadata is extracted from the HTML. The advantage of using this approach is twofold: (i) the publication data used is up-to-date and includes recent publications as well as authors; and, (ii) it does not require too much computer resources to handle and parse the entire dump of a static dataset. Thereafter, the local database is populated with additional information (affiliation and e-mail IDs, where available) from the Web in a resource bound manner following a [Pereira et al. 2009].

**Stage 1**

4.4.2.1.1 **Clusters based on E-mail information**

In this, we use e-mail ID(s) in the publications to group them into clusters. We used concepts of graph theory for this following [Liu et al., 2014b], where they were used to create co-authors based clusters. For example, if we have five citation-records in the set \( CR = \{cr_1, cr_2, cr_3, cr_4, cr_5\} \), we define a bit-matrix (BE) of order \( n \times n \) (where \( n \) is the number of citation-records in \( CR \)) to express the relationships between citation-records in \( CR \) using Equation 4.22.

\[
cr_{mn} = \begin{cases} 1, & \text{if } cr_m \text{ and } cr_n \text{ have at least one matching email ID} \\ 0, & \text{otherwise} \end{cases}
\]  

(4.22)
Suppose we have the following bit-matrix $BE$ for the set $CR$:

$$BE = \begin{bmatrix}
-1 & -1 & -1 & -1 & -1 \\
1 & -1 & -1 & -1 & -1 \\
0 & 0 & -1 & -1 & -1 \\
1 & 0 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & -1 \\
\end{bmatrix} \tag{4.23}$$

The value -1 for a particular pair $(m, n)$ in (4.23) indicates that we don’t need to compare $cr^m$ and $cr^n$. This reduces unnecessary comparisons. It follows from (4.19) that the following sets of citation-records have matching e-mail IDs: $(cr^2, cr^4), (cr^4, cr^3), (cr^4, cr^3)$. This can be represented as a graph as in Figure 4.8. This figure shows that there are two e-mail ID based clusters \{cr^1, cr^2, cr^3, cr^4\} and \{cr^5\}. Using Floyd-Warshall algorithm [Cormen et al., 2001; Aini and Salehipour, 2012] the bit-matrix shown in Equation 4.23 can be converted into a reachability matrix $RE$ by calculating the distance between citation-records. A distance of 0 in the reachability matrix indicates absence of a common e-mail ID between two citation-records whereas 1 indicates that they have a common e-mail ID.

![Figure 4.7: Relationship between citation-records based on e-mail information](image-url)
Figure 4.7 depicts the relationship between citation-records on the basis of matching e-mail IDs. Citation-records which share a common e-mail ID are merged in this stage of clustering.

The reachability matrix $RE$ for e-mail similarity between citation-records for the above example is:

$$RE = \begin{bmatrix}
    cr^1 & cr^2 & cr^3 & cr^4 & cr^5 \\
    cr^1 & 1 & 1 & 1 & 0 \\
    cr^2 & 1 & 1 & 1 & 0 \\
    cr^3 & 0 & 1 & 1 & 1 & 0 \\
    cr^4 & 1 & 1 & 1 & 1 & 0 \\
    cr^5 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \quad (4.24)$$

### 4.4.2.1.2 Clusters Based on Affiliation Information

In Chapter-3, we presented the results of our proposed affiliation similarity computation mechanism. We used a mix of Jaro-Winkler and Cosine similarities to determine the level of similarity between two affiliation strings. The similarity is either 0 or 1 and is calculated using Equation 4.25.

$$AffSim \begin{cases} 
    1, & \text{if JaroWinkler Similarity } \geq \xi \\
    1, & \text{if CosineSimilarity } \geq \beta \text{ and JaroWinkler Similarity } \geq \varepsilon \\
    1, & \text{if CosineSimilarity } \geq \delta \text{ and JaroWinkler Similarity } \geq \zeta \\
    0, & \text{otherwise}
\end{cases} \quad (4.25)$$

Here, $\xi$, $\beta$, $\varepsilon$, $\delta$ and $\zeta$ are thresholds. For string comparison both JaroWinkler and Cosine similarity return values in the range [0, 1]. After various test runs, we got best results with $\xi=0.79$, $\beta=0.56$, $\varepsilon=0.66$, $\delta=0.59$ and $\zeta=0.6$. In order to check the efficiency of the proposed affiliation matching approach we tested it against affiliation strings used in previous studies also. Results thereof have been already presented in Section 3.4 of Chapter-3.

Broad contours of the procedure used to find affiliation similarity are presented in Procedure-1.


**Procedure-1**: Affiliation-based Clusters

| Input: email ID-based Intermediate Results ($R_i$) |
| Output: Affiliation-based Intermediate Results ($R_a$) |

1. for each citation record $cr_i$ in cluster $c_k$ do
2.   for each citation record $cr_j$ in cluster $c_l$ ($k \neq l$ and $l = k + 1$) do
3.     if AffSim ($T (cr_i, cr_j)$) > threshold then
4.       Merge $c_l$ with $c_k$ and remove $c_l$ from $R_i$;
5.     end if
6.   end for
7. end for
8. $R_a = R_i$

---

**4.4.2.1.3 Clusters Based on Co-author Information**

The third step in the agglomerative clustering process is merging clusters generated in the second step on the basis of their sharing a co-author between them. The pairwise similarity on the basis of their sharing a co-author between any two citation-records $cr_i$ and $cr_j$, where $cr_i$ and $cr_j$ are part of two different clusters, on the basis of their sharing a co-author is calculated in two steps. Firstly their cosine similarity is calculated and if it is above a threshold, then JaroWinkler similarity is calculated. The calculation is as in Equation 4.26.

\[
Co - AuthSim = \begin{cases} 
1, & \text{if } CosineSimilarity \geq \beta \text{ and } JaroWinklerSimilarity \geq \xi \\
0, & \text{otherwise} 
\end{cases} \tag{4.26}
\]

In our experiments the values of two thresholds $\beta$ and $\xi$ used in (4.26) are set at 0.5 and 0.8 respectively. The process can be demonstrated using the following example.

Suppose we have two clusters $C_i$ and $C_j$ having citation-records $\{cr_{i1}, cr_{i2} \text{ and } cr_{i3}\}$ and $\{cr_{j1}, cr_{j2}\}$ respectively. These two clusters $C_i$ and $C_j$ are merged to create a new cluster $C_k$ if we find matching co-authors between any of
these pairs \((cr_{i1}, cr_{j1}), (cr_{i1}, cr_{j2}), (cr_{i2}, cr_{j1}), (cr_{i3}, cr_{j1})\) and \((cr_{i3}, cr_{j2})\), stopping when a matching co-author is found between citation-records in any of these pairs. This is illustrated in Figure 4.8. In this example matching co-author is found between \((cr_{i1}, cr_{j2})\), i.e. \(ca_{i12}\) and \(ca_{j22}\) refer to the same co-author and there is no need to compare the remaining pairs i.e. \((cr_{i2}, cr_{j1}), (cr_{i2}, cr_{j2}), (cr_{i3}, cr_{j1})\) and \((cr_{i3}, cr_{j2})\).

![Figure 4.8: Co-author based merging example.](image)

The co-author based merging process is outlined in Procedure 2.

**Procedure-2: Co-author-based Clusters**

**Input:** Affiliation-based Intermediate Results \((R^a)\)

**Output:** Co-author-based Intermediate Results \((R^c)\)

1. \(\text{for each citation record } cr_i \text{ in cluster } c_k \text{ do}\)
2. \(\quad \text{for each citation record } cr_j \text{ in cluster } c_l \text{ (} k \neq l \text{ and } l=k+1 \text{) do}\)
3. \(\quad \quad \text{if Co-AuthorSim } (\mathcal{F}(cr_i, cr_j)) > \text{threshold then}\)
4. \(\quad \quad \quad \text{Merge } c_i \text{ with } c_k \text{ and remove } c_i \text{ from } R^a;\)
5. \(\quad \quad \text{Go to next cluster pair}\)
6. \(\quad \text{end if}\)
7. \(\quad \text{end for}\)
8. \(\text{end for}\)
9. \(R^c \leftarrow R^a\)
4.4.2.1.4 Clusters Based on Publication-venue Information

The similarity between publication venue details is calculated using Equation 4.27 as:

\[
\text{VenueSim} = \begin{cases} 
1, & \text{if } \text{CosineSimilarity} > \zeta \text{ and JaroWinklerSimilarity} > \varphi \\
0, & \text{otherwise}
\end{cases} \quad (4.27)
\]

The values of threshold \( \zeta \) and \( \varphi \) in our experiments are 0.49 and 0.8 respectively. Procedure-3 outlines the publication-venue based merging process.

Procedure-3: Venue-based Clusters

**Input:** Co-author-based Intermediate Results \((R^c_i)\)

**Output:** Venue-based final Results \((R_f)\)

1. **for** each citation record \(cr_i\) in cluster \(c_k\) **do**
2. **for** each citation record \(cr_j\) in cluster \(c_l\) \((k \neq l \text{ and } l=k+1)\) **do**
3. **if** Co-AuthSim \((F (cr_i, cr_j)) \) > threshold **then**
4. Merge \(c_l\) with \(c_k\) and remove \(c_l\) from \(R^c_i\);
5. **Go to** next cluster pair
6. **end if**
7. **end for**
8. **end for**
9. \(R_f \leftarrow R^c_i\)

Stage 2

4.4.2.1.5 Clusters Based on Fuzzy Membership Value

As discussed earlier in this chapter, fuzzy or soft clustering allows data elements to belong to more than one cluster simultaneously and be associated with each of these clusters with certain membership levels. The membership level can be quantified in terms of degree of membership or membership values. The degree or grade of membership which can be any value in the range \([0, 1]\) indicates the strength of association between that
data element and a particular cluster. Soft clustering is a process of assigning these membership values, and then using them to assign data elements to one or more clusters [Wang et al., 2011]. Soft clustering is beneficial in dealing with uncertainty. There may be certain cases where agglomerative clustering used in first stage may have a good number of clusters with only one citation record. In such cases we use fuzzy clustering to find the relative similarity between singleton clusters and the rest of the clusters by calculating the value of membership function (μ) using Equation 4.28.

$$
\mu_{ij}(cr_i, C_j) = \frac{\cos(cr_i, C_j)^m}{\sum_{t=1}^{k} \cos(cr_i, C_t)^m}
$$

(4.28)

where $cr_i$ is the $i^{th}$ publication in a singleton cluster and $C_j$ is the $j^{th}$ cluster ($i \neq j$), respectively, and $m$ is the fuzzy factor.

The parameter $m$ determines the “softness” of the clustering solution. If $m=0$, the degree of membership of a publication with all the remaining clusters is same and when $m$ approaches $\infty$, the clustering becomes hard clustering [Zhao and Karypis, 2004]. In general, the softness of the clustering solution is inversely proportional to fuzzy factor $m$. In our case we merge a singleton cluster with any other cluster only if the value of the fuzzy membership function is above a threshold.

The value of $\mu_{ij}(cr_i, C_j)$ decides whether the two clusters can be merged or not. The similarity between two such clusters can be defined using Equation 4.29.

$$
FuzzSim = \begin{cases} 
1, & \text{if } (SCP \leq 30) \text{ and } (\mu_{ij} \geq 0.5) \\
1, & \text{if } (SCP > 30 \text{ and } SCP \leq 60) \text{ and } (\mu_{ij} \geq 0.3) \\
1, & \text{if } (SCP > 60) \text{ and } (\mu_{ij} \geq 0.2) \\
0, & \text{otherwise}
\end{cases}
$$

(4.29)
where, SCP is percentage of singleton clusters generated in step-6 of the proposed name disambiguation algorithm (see Figure 4.5) and $\mu$ is value of membership function.

The fuzzy membership value based merging process is outlined in Procedure 4.

**Procedure-4: Fuzzy Membership Value-based Clusters**

| Input: 6th Step Intermediate Results ($R_i^p$) |
| Output: Final Results ($R_f$) |
| 1. Find the number of singleton clusters and their percentage |
| 2. for each singleton cluster $c_j$ do |
| 3. for each non-singleton cluster $c_k$ do |
| 4. Calculate the value of membership ($\mu$) of $c_j$ with $c_k$ |
| 5. if $\mu_{jk} > $ threshold then |
| 6. Merge $c_j$ with $c_k$ and remove $c_j$ from $R_i^p$; |
| 7. Go to next cluster pair |
| 8. end if |
| 9. end for |
| 10. end for |
| 11. $R_f \leftarrow R_i^p$ |

### 4.5 Results and Discussion

#### 4.5.1 Experimental Setup

We attempted to disambiguate a number of ambiguous author groups using the attributes principal author, co-authors, e-mail IDs of principal author and co-authors, affiliations, publication title, publication venue and year of publication. After extraction from their respective sources these attributes were stored in a database. Instead of combining all these attributes, in the first stage the clusters were generated on the basis of an attribute suitable for that step of clustering. All the attributes were used for this one after another. In the second stage i.e. fuzzy clustering stage, the similarity score was computed on the basis of all the available attributes.
4.5.2 Evaluation Parameters

Bonner [Bonner, 1964], terms the efficiency of clustering methods as a problematic and controversial issue and argues that there is no universal definition or parameter, that fits all situations, for answering *how good a clustering technique is?* Therefore, the quality of clusters is largely defined in terms of user satisfaction and domain applicability. A number of such evaluation criteria have been proposed in literature. Rokach et al. [Rokach and Maimon, 2005] group them into two classes: Internal Quality Criteria and External Quality Criteria. The compactness of the clusters, intra-cluster similarity and inter-cluster distance, are determined by using internal quality metrics, whereas the similarity between generated and expected clusters is determined using external quality criteria [Rokach and Maimon, 2005].

We use the traditional *precision-recall* measure from information retrieval to measure the quality of the generated clusters. In our case, *precision* refers to the fraction of correctly clustered publications, while *recall* refers to the fraction of correctly clustered publications out of all actual publications of the same author. *Precision, Recall* and *F-1 score* are calculated using Equations 4.30, 4.31 and 4.32, respectively, as used in [Tang et al., 2012; Liu et al., 2014b], as follows:

\[
\text{Precision} = \frac{\text{Citation Records Correctly Predicted to Same Author}}{\text{Total Citation Records Predicted to Same Author}} \tag{4.30}
\]

\[
\text{Recall} = \frac{\text{Citation Records Correctly Predicted to Same Author}}{\text{Total Citation Records of Same Author}} \tag{4.31}
\]

\[
F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.32}
\]

4.5.3 Dataset

The dataset used for author name disambiguation in this work was obtained from DBLP. DBLP has been used in similar studies a number of
times in the past and has become a benchmark source for datasets for name disambiguation techniques. Countless number of authors are listed in DBLP and other scientific literature indexing services. So it may not be possible to include data for all these authors in any single study. It may not be possible even to include data for all the authors indexed by a single indexing or digital library in one study, as has been discussed in Chapter 3. The size and complexity of DBLP can be estimated from the fact that as on January 20, 2015 DBLP has indexed 28,58,971 publications\(^ {20} \). The breakup of these publications is shown in Figure 4.9.

In this background, it becomes difficult to decide which authors to consider for testing the efficiency of the proposed disambiguation approach. In order to maintain a balance between data size and complexity on the one hand and resource demand on the other, we decided to test the proposed approach against ambiguous author names commonly used in some of the previous studies [Wang et al., 2008; Wang et al., 2011; Tang et al., 2012; Imran et al., 2013; Arif et al., 2014c]. Table 4.2 lists the values of various attributes of the dataset.

![Figure 4.9: Distribution of DBLP records by publication type [Source: DBLP].](http://www.informatik.uni-trier.de/~ley/statistics/recordsindblp.html)

\(^ {20} \) http://www.informatik.uni-trier.de/~ley/statistics/recordsindblp.html
**Table 4.2: Values of various attributes of the dataset.**

<table>
<thead>
<tr>
<th>Author</th>
<th>Publications</th>
<th>#Actual Authors</th>
<th>Author</th>
<th>Publications</th>
<th>#Actual Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Smith</td>
<td>81</td>
<td>6</td>
<td>Alok Gupta</td>
<td>94</td>
<td>5</td>
</tr>
<tr>
<td>Ajay Gupta</td>
<td>107</td>
<td>11</td>
<td>David Jensen</td>
<td>82</td>
<td>5</td>
</tr>
<tr>
<td>Michael Wagner</td>
<td>232</td>
<td>20</td>
<td>Charles Smith</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>Rakesh Kumar</td>
<td>233</td>
<td>33</td>
<td>Ajay Kumar</td>
<td>45</td>
<td>8</td>
</tr>
<tr>
<td>Hui Fang</td>
<td>156</td>
<td>21</td>
<td>Rashid Ali</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>Jie Tang</td>
<td>227</td>
<td>12</td>
<td>Robert Moore</td>
<td>91</td>
<td>12</td>
</tr>
<tr>
<td>Richard Taylor</td>
<td>186</td>
<td>19</td>
<td>William Cohen</td>
<td>190</td>
<td>5</td>
</tr>
<tr>
<td>Paul Jones</td>
<td>57</td>
<td>11</td>
<td>Joseph Miller</td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>Robert Fisher</td>
<td>191</td>
<td>11</td>
<td>Dmitry Pavlov</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td>Gang Wu</td>
<td>295</td>
<td>24</td>
<td>Robert Williams</td>
<td>56</td>
<td>15</td>
</tr>
</tbody>
</table>

**4.5.4 Name Disambiguation Results**

The results of our experimentation for author name disambiguation with the proposed approach are presented in Tables 4.3 and 4.4. Table 4.3 presents the values returned for number of auto-authors, i.e. number of different authors with the same name, as generated/recognized by our algorithm. The actual number of different authors with the same name is also shown in these tables for comparison. Besides, it shows the values for precision, recall and F1 obtained without fuzzy clustering step.
Table 4.3.: Author Name Disambiguation Results Obtained using the Proposed Approach – OSMC (without Fuzzy Clustering).

<table>
<thead>
<tr>
<th>Author</th>
<th>#Actual-Autors</th>
<th>#Auto-Autors</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Smith</td>
<td>6</td>
<td>6</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Ajay Gupta</td>
<td>11</td>
<td>14</td>
<td>97.09</td>
<td>96.15</td>
<td>96.62</td>
</tr>
<tr>
<td>Michael Wagner</td>
<td>20</td>
<td>25</td>
<td>96.70</td>
<td>90.71</td>
<td>93.61</td>
</tr>
<tr>
<td>Rakesh Kumar</td>
<td>33</td>
<td>31</td>
<td>83.64</td>
<td>90.40</td>
<td>86.89</td>
</tr>
<tr>
<td>Hui Fang</td>
<td>21</td>
<td>25</td>
<td>97.39</td>
<td>98.03</td>
<td>97.70</td>
</tr>
<tr>
<td>Jie Tang</td>
<td>12</td>
<td>13</td>
<td>97.80</td>
<td>100.00</td>
<td>98.89</td>
</tr>
<tr>
<td>Richard Taylor</td>
<td>19</td>
<td>24</td>
<td>89.89</td>
<td>95.24</td>
<td>92.49</td>
</tr>
<tr>
<td>Paul Jones</td>
<td>11</td>
<td>13</td>
<td>88.46</td>
<td>90.20</td>
<td>89.32</td>
</tr>
<tr>
<td>Robert Fisher</td>
<td>11</td>
<td>17</td>
<td>95.29</td>
<td>100.00</td>
<td>97.59</td>
</tr>
<tr>
<td>Gang Wu</td>
<td>24</td>
<td>40</td>
<td>87.50</td>
<td>54.65</td>
<td>67.28</td>
</tr>
<tr>
<td>Alok Gupta</td>
<td>5</td>
<td>6</td>
<td>98.94</td>
<td>100.00</td>
<td>99.47</td>
</tr>
<tr>
<td>David Jensen</td>
<td>5</td>
<td>5</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Charles Smith</td>
<td>15</td>
<td>16</td>
<td>97.50</td>
<td>100.00</td>
<td>98.73</td>
</tr>
<tr>
<td>Ajay Kumar</td>
<td>8</td>
<td>7</td>
<td>100.00</td>
<td>95.56</td>
<td>97.73</td>
</tr>
<tr>
<td>Rashid Ali</td>
<td>5</td>
<td>5</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Robert Moore</td>
<td>12</td>
<td>18</td>
<td>91.21</td>
<td>100.00</td>
<td>95.40</td>
</tr>
<tr>
<td>William Cohen</td>
<td>5</td>
<td>7</td>
<td>98.95</td>
<td>100.00</td>
<td>99.47</td>
</tr>
<tr>
<td>Joseph Miller</td>
<td>4</td>
<td>6</td>
<td>89.29</td>
<td>100.00</td>
<td>94.34</td>
</tr>
<tr>
<td>Dmitry Pavlov</td>
<td>3</td>
<td>4</td>
<td>96.30</td>
<td>100.00</td>
<td>98.11</td>
</tr>
<tr>
<td>Robert Williams</td>
<td>15</td>
<td>24</td>
<td>87.04</td>
<td>95.92</td>
<td>91.26</td>
</tr>
</tbody>
</table>

| Average         | 94.65          | 95.34        | 94.74     |

Table 4.4 presents the final results of the clustering process influenced by the concept of fuzzy clustering.
Chapter 4: Metadata-based Author Name Disambiguation & Profile Integration

Table 4.4.: Author Name Disambiguation Results Obtained by the Proposed Approach – TSMC (with Fuzzy Clustering).

<table>
<thead>
<tr>
<th>Author</th>
<th>#Actual-Authors</th>
<th>#Auto-Authors</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Smith</td>
<td>6</td>
<td>6</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Ajay Gupta</td>
<td>11</td>
<td>13</td>
<td>97.12</td>
<td>96.19</td>
<td>96.65</td>
</tr>
<tr>
<td>Michael Wagner</td>
<td>20</td>
<td>19</td>
<td>100.00</td>
<td>95.50</td>
<td>97.70</td>
</tr>
<tr>
<td>Rakesh Kumar</td>
<td>33</td>
<td>28</td>
<td>84.51</td>
<td>89.11</td>
<td>86.75</td>
</tr>
<tr>
<td>Hui Fang</td>
<td>21</td>
<td>23</td>
<td>98.04</td>
<td>97.40</td>
<td>97.72</td>
</tr>
<tr>
<td>Jie Tang</td>
<td>12</td>
<td>12</td>
<td>98.23</td>
<td>99.55</td>
<td>98.89</td>
</tr>
<tr>
<td>Richard Taylor</td>
<td>19</td>
<td>20</td>
<td>90.45</td>
<td>93.60</td>
<td>92.00</td>
</tr>
<tr>
<td>Paul Jones</td>
<td>11</td>
<td>13</td>
<td>88.46</td>
<td>90.20</td>
<td>89.32</td>
</tr>
<tr>
<td>Robert Fisher</td>
<td>11</td>
<td>16</td>
<td>95.81</td>
<td>100.00</td>
<td>97.86</td>
</tr>
<tr>
<td>Gang Wu</td>
<td>24</td>
<td>37</td>
<td>89.09</td>
<td>54.04</td>
<td>67.28</td>
</tr>
<tr>
<td>Alok Gupta</td>
<td>5</td>
<td>5</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>David Jensen</td>
<td>5</td>
<td>5</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Charles Smith</td>
<td>15</td>
<td>16</td>
<td>97.50</td>
<td>100.00</td>
<td>98.73</td>
</tr>
<tr>
<td>Ajay Kumar</td>
<td>8</td>
<td>7</td>
<td>100.00</td>
<td>95.56</td>
<td>97.73</td>
</tr>
<tr>
<td>Rashid Ali</td>
<td>5</td>
<td>5</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Robert Moore</td>
<td>12</td>
<td>18</td>
<td>91.21</td>
<td>100.00</td>
<td>95.40</td>
</tr>
<tr>
<td>William Cohen</td>
<td>5</td>
<td>7</td>
<td>98.95</td>
<td>100.00</td>
<td>99.47</td>
</tr>
<tr>
<td>Joseph Miller</td>
<td>4</td>
<td>4</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Dmitry Pavlov</td>
<td>3</td>
<td>3</td>
<td>100.00</td>
<td>100.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Robert Williams</td>
<td>15</td>
<td>22</td>
<td>90.74</td>
<td>96.08</td>
<td>93.33</td>
</tr>
</tbody>
</table>

Average   96.01  95.36  95.68

4.5.4.1 Analysis of Results

To compare the prediction efficiency of the proposed approach we compared our results with those obtained in [Tang et al., 2012] using Fixed-K. Like the authors in [Tang et al., 2012] we also compared our results with those obtained using: (i) HAC [Tan et al., 2006] which uses agglomerative clustering and augments the metadata using search engine results; (ii) CONSTRAINT [Zhang et al., 2007], a constraint-based clustering algorithm. Table 4.5 lists the values of precision, recall and F1 scores listed in [Tang et
al., 2012] for HAC, CONSTRAINT and Fixed-K in comparison with those returned by our proposed approach both without fuzzy clustering and with fuzzy clustering. Disambiguation results without fuzzy clustering are listed under One Stage Multiple Clustering (OSMC) column and those with fuzzy clustering are listed under Two Stage Multiple Clustering (TSMC).

For a better comparison of these techniques we compared the values of the three evaluation metrics for each of them and presented the comparative scores in Figures 4.10, 4.11 and 4.12. Figure 4.10 presents the comparison on the basis of precision, 4.11 on the basis of recall and 4.12 on the basis of F1.

4.5.4.1.1 Successful Cases

From the perusal of Tables 4.4 and 4.5, it is evident that the proposed approach has been able to disambiguate ambiguous authors to a great extent. The cumulative $F1$ score returned by our approach for all the ambiguous authors listed in Table 4.5 is much higher than that obtained using the techniques followed in [Tan et al., 2006; Zhang et al., 2007; Tang et al., 2012]. These results are also an improvement over those of [Wang et al., 2008; Wang et al., 2011; Imran et al., 2013]. From Table 4.5, it can be observed that HAC and CONSTRAINT were able to achieve hundred percent $F1$ in case of two author names each, and Fixed-K achieved that in case of three author names. Our proposed approach achieved hundred percent $F1$ score in case of four author names.

Of the twenty author names listed in Tables 4.2 and 4.4, the proposed approach successfully placed publications of six authors in desired number of clusters i.e. the number of authors generated was equal to number of actual authors. These author names are Jim Smith, Alok Gupta, David Jensen, Rashid Ali, Joseph Miller and Dmitry Pavlov. The results for these authors suffer neither from Split Citation problem nor from Mixed Citation problem. In all these cases the values of precision, recall as well as $F1$ were 100%. The value of $F1$ was more than 95% in case of nine ambiguous author
names, 90-95% in case of two, 85-90% in case of two and below 85% in case of only one author name.

The proposed method was able to deal with mixed-citations problem very efficiently as the value of recall was 100% in case of ten author names. The value of recall was more than 95% in case of six author names, 90-95% in case of two, 85-90% in case of one and below 85% in case of only one.

The proposed method was able to deal with split-citations problems also very efficiently as the value of precision was 100% in case of eight author names. The values of recall were more than 95% in case of six author names, 90-95% in case of three, 85-90% in case of two and below 85% in case of one only.

4.5.4.1.2 Failed Cases

The values of all the three evaluation metrics viz. precision, recall and $F_1$ returned by our approach are very good in majority of the cases. But in case of Gang Wu the values of precision, recall and $F_1$ are 89.09, 54.04 and 67.28, respectively. The low value of $F_1$ in this case is due to low value of recall, which in turn means that this particular name faced mixed-citations problem. During experimentation it was observed that more than one authors with the name Gang Wu published either in the same journals or in the same conference proceedings. This led to a rise in false negatives and consequently low value of recall.

In other cases, the comparatively low value of recall was found due to merging of publications of different actual authors during the venue-based merging phase.
Table 4.5: Comparison of Precision, Recall and F1 Values of Various Name Disambiguation Methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>HAC</th>
<th>CONSTRAINT</th>
<th>Fixed-K</th>
<th>OSMC (Without Fuzzy Clustering)</th>
<th>TSMC (With Fuzzy Clustering)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Smith</td>
<td>92.43</td>
<td>86.80</td>
<td>89.53</td>
<td>70.91</td>
<td>97.50</td>
</tr>
<tr>
<td>Ajay Gupta</td>
<td>41.88</td>
<td>100.0</td>
<td>59.04</td>
<td>90.67</td>
<td>96.55</td>
</tr>
<tr>
<td>Michael Wagner</td>
<td>39.35</td>
<td>60.26</td>
<td>79.53</td>
<td>26.25</td>
<td>77.78</td>
</tr>
<tr>
<td>Rakesh Kumar</td>
<td>63.36</td>
<td>92.41</td>
<td>75.18</td>
<td>92.37</td>
<td>99.18</td>
</tr>
<tr>
<td>Hun Fang</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Jie Tang</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Richard Taylor</td>
<td>80.17</td>
<td>90.93</td>
<td>88.97</td>
<td>68.23</td>
<td>64.54</td>
</tr>
<tr>
<td>Paul Jones</td>
<td>36.36</td>
<td>80.00</td>
<td>50.00</td>
<td>38.64</td>
<td>63.45</td>
</tr>
<tr>
<td>Robert Fisher</td>
<td>96.14</td>
<td>100.0</td>
<td>98.03</td>
<td>85.21</td>
<td>74.54</td>
</tr>
<tr>
<td>Gang Wu</td>
<td>97.54</td>
<td>97.54</td>
<td>97.54</td>
<td>71.86</td>
<td>98.36</td>
</tr>
<tr>
<td>David Jensen</td>
<td>85.85</td>
<td>94.88</td>
<td>90.14</td>
<td>82.54</td>
<td>65.23</td>
</tr>
<tr>
<td>Charles Smith</td>
<td>30.00</td>
<td>100.0</td>
<td>46.15</td>
<td>45.27</td>
<td>67.89</td>
</tr>
<tr>
<td>Robert Moore</td>
<td>86.90</td>
<td>93.10</td>
<td>89.89</td>
<td>89.91</td>
<td>78.54</td>
</tr>
<tr>
<td>William Cohen</td>
<td>81.53</td>
<td>97.89</td>
<td>89.00</td>
<td>86.94</td>
<td>85.23</td>
</tr>
<tr>
<td>Joseph Miller</td>
<td>54.55</td>
<td>54.55</td>
<td>54.55</td>
<td>55.21</td>
<td>59.34</td>
</tr>
<tr>
<td>Dmitry Pavlov</td>
<td>85.71</td>
<td>85.71</td>
<td>85.71</td>
<td>88.70</td>
<td>89.23</td>
</tr>
<tr>
<td>Robert Williams</td>
<td>66.67</td>
<td>66.67</td>
<td>66.67</td>
<td>65.12</td>
<td>58.23</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>71.81</td>
<td>88.81</td>
<td>76.97</td>
<td>73.93</td>
<td>80.92</td>
</tr>
</tbody>
</table>

Pr. & Rec. refer to Precision and Recall, respectively.
Figure 4.10: Comparative values of Precision.

Figure 4.11: Comparative values of Recall.
It is important for organizations and interested parties, like funding agencies, to have a fairly accurate idea of the research output of a person. Funding agencies need this information to evaluate the research potential of a candidate for funding, for finding appropriate people/research groups to start a new research project, etc. The profile of an author/researcher provides a crisp overview of his research credentials. Among the elements of a profile, the work published by a researcher is the most important indicator of his potential. This fact has been amply brought out by a number of studies [Luukkonen et al., 1992; Georghiou, 1998; Gla’nzel, 2001; Gla’nzel, 2002; Zitt, and Bassecoulard, 2004; Wagner and Leydesdorff, 2005a; Wagner and Leydesdorff, 2005b; Wagner and Leydesdorff, 2008; Vidican et al., 2009; Chang and Huang, 2014], that have used publications data of authors for understanding a number of research related facts about them.
We make use of the disambiguated publication-records of an author to integrate his profile. Since publications data is a rich source of information on a researcher, we extract some of its critical elements like number and pattern of research publications and research collaborations, affiliations and shifts therein, etc. This publication and profile information is used for extraction and analysis of academic social networks in Chapter-5.

In this case, we are interested in defining the profile of an author on the basis of some of the features extracted from the disambiguated data. These features include the total number of publications (#TotalPublications), affiliations (Works/WorkedFor), co-authors (Co-Authors), publication frequency with each of the co-authors (#PapersWith) and recently published papers in (RecentVenue).

The proposed algorithm for author profile integration is shown in Figure 4.13.

Table 4.6 presents the results obtained by the profile integration algorithm discussed in Figure 4.13 for the author name ‘Rashid Ali’. The profile attributes for which statistics are listed in this table provide a snapshot of the research and work profile for this author. The same profile integration procedure can be applied to perform profile integration for other authors too.

Analysis of various fields in Table 4.6 provides a glimpse of important facts about an author. Co-Authors field provides a list of the people with which the Target-Author collaborates for joint publications, the more the number of persons in this list, the more collaborative the author is. The field #PapersWith provides the frequency with which the author publishes jointly with his co-authors. These statistics shall be used for visualization and analysis of important social network metrics in Chapter-5.
**Algorithm:** Author Profile Integration

**Input:** Target-Auth Account Name and Disambiguated Publications Data

**Output:** Profile of Given Target-Auth Account Name

1. [Extract $\#TotalPublications$] Extract total number of publications from the cluster for the current author.

2. [Extract Works/WorkedFor]: Extract and assign to a list all affiliations of Target Author from Disambiguated Publications Clusters.

3. [Remove Duplicates from Works/WorkedFor] Find and remove matching co-authors from the list generated in Step-2 using fuzzy string matching techniques (*Cosine Similarity* in 1st phase and *Jaro-Winkler Similarity* in 2nd phase, if CosineSimilarity > threshold).

4. [Extract Co-Authors] Extract all the Co-authors of Target Author from Disambiguated Publications Clusters.

5. [Remove Duplicates from Co-Authors] Find and remove matching co-authors from the co-author list generated in Step-4 using fuzzy string matching techniques (*Cosine Similarity* in 1st phase and *Jaro-Winkler Similarity* in 2nd phase, if CosineSimilarity > threshold).

6. [Extract $\#PapersWith$] Extract the number of publications of the Target Author with each of the Co-Authors generated in Step-5.

7. [Extract RecentVenue] Extract the venue of the conference or name of the journal in which the Target Author has published most recently.

---

Figure 4.13: Algorithm for Profile Integration
Table 4.6: Profiles & Statistics for Author-Name ‘Rashid Ali’.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Total Publications</td>
<td>23</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Works/WorkedFor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. College of Computers &amp; IT, Taif University, Saudi Arabia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Department of Computer Engineering, A. M. U., Aligarh, 20202, India</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Department of Computer Science Cardiff University, UK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vishveshwarya Institute of Engineering and Technology, Padri, G. B. Nagar, U.P., India</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division of Biomedical Informatics, Sidsa Medical and Research Centre, Qatar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Department of Biochemistry, Faculty of Medicine, Jawaharlal Nehru Medical College, Aligarh Muslim University, Aligarh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Co-Authors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Majid Bahar Malik</td>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
</tr>
<tr>
<td>M. Asgar</td>
<td>Tareen Arif</td>
<td>M. Usman</td>
<td>Shaheyl SJ</td>
<td>Tanveer Shai</td>
<td>Miroslova Sarefka</td>
</tr>
<tr>
<td>Mirza Mohd Safiyan Beg</td>
<td>Shahab Siddiqui</td>
<td>Manzoor Ahmad</td>
<td>Kauser Amin</td>
<td>Gregor Von Laszewski</td>
<td>Omar E. Rana</td>
</tr>
<tr>
<td>Samir Jha</td>
<td>Sanjoy Jha</td>
<td>Bani Singh</td>
<td>Anuj Bhardwaj</td>
<td>Abdulhakim Hafez</td>
<td>David W. Walker</td>
</tr>
<tr>
<td>Tasleem Arif</td>
<td>Mihael Hategan</td>
<td>Jamshed Siddiqui</td>
<td>Supriya Kamthania</td>
<td>Om Prakash</td>
<td>M. Asgar</td>
</tr>
<tr>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
<td>M. Usman</td>
</tr>
<tr>
<td>Tareen Arif</td>
<td>M. Usman</td>
<td>Shaheyl SJ</td>
<td>Tanveer Shai</td>
<td>Miroslova Sarefka</td>
<td></td>
</tr>
<tr>
<td>Miroslova Sarefka</td>
<td>Shahab Siddiqui</td>
<td>Manzoor Ahmad</td>
<td>Kauser Amin</td>
<td>Gregor Von Laszewski</td>
<td>Omar E. Rana</td>
</tr>
<tr>
<td>Samir Jha</td>
<td>Sanjoy Jha</td>
<td>Bani Singh</td>
<td>Anuj Bhardwaj</td>
<td>Abdulhakim Hafez</td>
<td>David W. Walker</td>
</tr>
<tr>
<td>Tasleem Arif</td>
<td>Mihael Hategan</td>
<td>Jamshed Siddiqui</td>
<td>Supriya Kamthania</td>
<td>Om Prakash</td>
<td>M. Asgar</td>
</tr>
<tr>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
<td>M. Usman</td>
</tr>
<tr>
<td>Tareen Arif</td>
<td>M. Usman</td>
<td>Shaheyl SJ</td>
<td>Tanveer Shai</td>
<td>Miroslova Sarefka</td>
<td></td>
</tr>
<tr>
<td>Miroslova Sarefka</td>
<td>Shahab Siddiqui</td>
<td>Manzoor Ahmad</td>
<td>Kauser Amin</td>
<td>Gregor Von Laszewski</td>
<td>Omar E. Rana</td>
</tr>
<tr>
<td>Samir Jha</td>
<td>Sanjoy Jha</td>
<td>Bani Singh</td>
<td>Anuj Bhardwaj</td>
<td>Abdulhakim Hafez</td>
<td>David W. Walker</td>
</tr>
<tr>
<td>Tasleem Arif</td>
<td>Mihael Hategan</td>
<td>Jamshed Siddiqui</td>
<td>Supriya Kamthania</td>
<td>Om Prakash</td>
<td>M. Asgar</td>
</tr>
<tr>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
<td>M. Usman</td>
</tr>
<tr>
<td>Tareen Arif</td>
<td>M. Usman</td>
<td>Shaheyl SJ</td>
<td>Tanveer Shai</td>
<td>Miroslova Sarefka</td>
<td></td>
</tr>
<tr>
<td>Miroslova Sarefka</td>
<td>Shahab Siddiqui</td>
<td>Manzoor Ahmad</td>
<td>Kauser Amin</td>
<td>Gregor Von Laszewski</td>
<td>Omar E. Rana</td>
</tr>
<tr>
<td>Samir Jha</td>
<td>Sanjoy Jha</td>
<td>Bani Singh</td>
<td>Anuj Bhardwaj</td>
<td>Abdulhakim Hafez</td>
<td>David W. Walker</td>
</tr>
<tr>
<td>Tasleem Arif</td>
<td>Mihael Hategan</td>
<td>Jamshed Siddiqui</td>
<td>Supriya Kamthania</td>
<td>Om Prakash</td>
<td>M. Asgar</td>
</tr>
<tr>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
<td>M. Usman</td>
</tr>
<tr>
<td>Tareen Arif</td>
<td>M. Usman</td>
<td>Shaheyl SJ</td>
<td>Tanveer Shai</td>
<td>Miroslova Sarefka</td>
<td></td>
</tr>
<tr>
<td>Miroslova Sarefka</td>
<td>Shahab Siddiqui</td>
<td>Manzoor Ahmad</td>
<td>Kauser Amin</td>
<td>Gregor Von Laszewski</td>
<td>Omar E. Rana</td>
</tr>
<tr>
<td>Samir Jha</td>
<td>Sanjoy Jha</td>
<td>Bani Singh</td>
<td>Anuj Bhardwaj</td>
<td>Abdulhakim Hafez</td>
<td>David W. Walker</td>
</tr>
<tr>
<td>Tasleem Arif</td>
<td>Mihael Hategan</td>
<td>Jamshed Siddiqui</td>
<td>Supriya Kamthania</td>
<td>Om Prakash</td>
<td>M. Asgar</td>
</tr>
<tr>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
<td>M. Usman</td>
</tr>
<tr>
<td>Tareen Arif</td>
<td>M. Usman</td>
<td>Shaheyl SJ</td>
<td>Tanveer Shai</td>
<td>Miroslova Sarefka</td>
<td></td>
</tr>
<tr>
<td>Miroslova Sarefka</td>
<td>Shahab Siddiqui</td>
<td>Manzoor Ahmad</td>
<td>Kauser Amin</td>
<td>Gregor Von Laszewski</td>
<td>Omar E. Rana</td>
</tr>
<tr>
<td>Samir Jha</td>
<td>Sanjoy Jha</td>
<td>Bani Singh</td>
<td>Anuj Bhardwaj</td>
<td>Abdulhakim Hafez</td>
<td>David W. Walker</td>
</tr>
<tr>
<td>Tasleem Arif</td>
<td>Mihael Hategan</td>
<td>Jamshed Siddiqui</td>
<td>Supriya Kamthania</td>
<td>Om Prakash</td>
<td>M. Asgar</td>
</tr>
<tr>
<td>Manzoor Ahmad Lone</td>
<td>S.M. Zakaria</td>
<td>Neeraj Ahmad</td>
<td>Om Prakash</td>
<td>Supritya Kamthania</td>
<td>M. Usman</td>
</tr>
</tbody>
</table>
4.7 Summary

Author Name Disambiguation is very important for proper attribution and assignment of credit among ambiguous authors. Effectiveness of any attempt to identify an author on the basis of his publication data or tabulated author profile is primarily dependent on name disambiguation technique used. The publications of an author contain a whole lot of explicit and implicit information about him like his co-authors, frequency and pattern of collaboration with co-authors, area of interest, preferred publication destinations (conferences/journals), most recent collaborations, venues and co-authors, etc. This information can be used for constructing profiles and social networks, and in turn helps in answering important questions about particular authors.

In this Chapter, we presented an approach for author name disambiguation. In terms of F1 scores, our method achieved improvement of 23.13% over HAC, 24.03% over CONSTRAINT and 8% over Fixed-K. We also observed that the results of disambiguation are better when a higher number of publication records are used.

Publication venue has been used for author name disambiguation in previous studies as well, but our experience with it indicates that with the ever increasing publications per venue, its usefulness for name disambiguation is on the wane as more and more similar authors are publishing in the same or relatively similar venues. The low recall and F1 score in case of Gang Wu is primarily due to the fact that more than one Gang Wu publishing in the same publication venues.

Profile integration is completely dependent on the results of the disambiguation process. A properly constructed profile conveys key information about an author at a glance, and is very useful for many interested stakeholders.