Disambiguating Authors in Bibliographic Databases

PhD Thesis

By
Muhammad Shoaib
26-FAS/PHDCS/S05

Supervisor: Dr. Ali Daud
Department of CS&SE, IIU, Islamabad

Co-supervisor: Prof. Dr. M. Sikander Hayat Khiyal
Faculty of CS, Preston University, Islamabad

Department of Computer Science & Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad, Pakistan
2015

International Islamic University, Islamabad
Faculty of Basic & Applied Sciences
Department of Computer Science

Dated:_____

FINAL APPROVAL

It is certified that we have read the thesis, entitled **"Disambiguating Authors in Bibliographic Databases"** submitted by Muhammad Shoaib, Reg. No. 26-FAS/PHDCS/S05. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for PhD Degree in Computer Science.

PROJECT EVALUATION COMMITTEE

External Examiners:
Prof. Dr. Sohail Asghar
Chief Technologist, Department of Computer Science
COMSATS Institute of Information Technology, Islamabad, Pakistan_____

Dr. Hasan Mujtaba Kayani
Associate Professor, Department of Computer Science
National University of Computer & Emerging Sciences, Islamabad, Pakistan_____

Internal Examiner:
Dr. Ayyaz Hussain
Assistant Professor, Department of CS & SE
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad, Pakistan _____

Supervisor:
Dr. Ali Daud
Assistant Professor, Department of CS & SE
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad, Pakistan _____

Co-Supervisor:
Prof. Dr. Malik Sikander Hayat Khiyal
Faculty of Computer Science
Preston University, Kohat, Islamabad Campus, Pakistan _____

Chairman
Dr. Hussnain Abbas Naqvi
Department of Computer CS & SE
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad, Pakistan _____

Dean
Prof. Dr. Muhammad Sher
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad, Pakistan _____

This dissertation is submitted to
International Islamic University Islamabad, Pakistan
In partial fulfillment of the requirement of the degree of
Doctor of Philosophy (Computer Science)

Acknowledgement

I would like to acknowledge the many people who have supported, guided and advised me during my PhD research work. First, I would like to express my sincere gratitude to my supervisor Dr. Ali Daud, Assistant Professor, Department of Computer Science & Software Engineering, International Islamic University, Islamabad; and co-supervisor Prof. Dr. Malik Sikandar Hayat Khiyal, Faculty of Computer Sciences, Preston University, Islamabad, for providing me an opportunity to start this study and research. Through the course of my studies, I have had the great fortune to get to know and interact with them. Their valuable comments, suggestions and motivations for further development as well as their assistance during writing PhD thesis are valuable to me. They always guided me whenever I stuck during the studies and encouraged during the whole period of my PhD degree.

I am also thankful to my friend, Nadeem Ahmad Shah, senior software engineer, Digital Processing Systems Inc., Islamabad, Pakistan who guided and helped me in implementing the proposed methodology.

I am thankful to all other persons who have directly or indirectly helped me in the completion of my work.

Declaration

I hereby declare and affirm that this thesis neither as a whole, nor as part thereof has been copied out from any source. It is further declared that I have completed this thesis entirely on the basis of my personal effort, made under the sincere guidance of my supervisors. If any part of this report is proven to be copied or found to be a reproduction of some other, I shall stand by the consequences. No portion of the work presented in this report has been submitted in support of an application for other degree or qualification of this or any other university or institute of learning.

Muhammad Shoaib

26-FAS/PHDCS/S05

Dedication

This work is dedicated to my parents, especially to my late mother who always prayed, encouraged, supported and guided me to achieve this milestone.

Muhammad Shoaib

Abstract

Author name disambiguation in bibliographic databases such as DBLP[1], Citeseer[2], and Scopus [3] is a specialized problem of entity resolution. In the literature, different approaches have been proposed and most of them base on machine learning techniques, either supervised or un-supervised learning or a combination of the two. The supervised learning approaches require labeling effort to train data. Unsupervised learning approaches utilize available attributes to group one's citations by exploiting different similarity measures and clustering algorithms. The performance of un-supervised methods is affected by clustering algorithms, attributes and similarity measures. Previously, the focus of the research was on devising clustering algorithms and identifying attributes, but similarity measures have not been paid due attention.

In this research work, we propose improved similarity measures for each type of attribute and a clustering algorithm. To estimate author name similarity, we divide name tokens into five different categories, and devise a similarity measure that accommodates them by assigning variant weights to each type of token. Our proposed similarity measure for co-authors attribute assigns higher similarity value to the citations if they share more common co-authors irrespective of the total number of co-authors. For textual attributes, we propose a conditional absolute measure (for attributes having short texts) and SDK[4] index (for attributes having long texts). Experiments on DBDComp datasets show that our similarity measures outperform baseline measures by 16.2% in k-measure and 14.20 % in f-measure.

We propose to use references of publications as additional sources of information. Use of titles of references improves k-measure by 0.6% and f-measure by 8% on DBLP-Ref datasets. We also propose clustering algorithm by modifying heuristic-based hierarchical clustering. Experiments on three different types of author name disambiguation collections show that our proposed methodology (similarity measures, clustering algorithm and use of references) helps improve both k-measure and f-measure.

[1] http://www.informatik.uni-trier.de/~ley/db/
[2] http://citeseer.ist.psu.edu/
[3] http://www.scopus.com/home.url
[4] Last names of th authors (Shoaib, Daud, Khiyal) who proposed this index

Achievements and Contribution

Journal Publications

1. M. Shoaib, A. Daud and M. S. H. Khiyal**, "**Improving similarity measures for publications with special focus on author name disambiguation", *Arabian Journal for Science and Engineering* (ISI IF journal), vol. 40, no. 6, pp: 1591-1605, 2015.
2. M. Shoaib, A. Daud, M. S. H. Khiyal, "Role of references in similarity estimation of publications." *The International Arab Journal of Information Technology* (ISI IF journal), in press.
3. M. Shoaib, A. Daud, M.S.H. Khiyal, "An improved similarity measure for text documents." *Journal of Basic and Applied Scientific Research* (ISI Thomson Reuters Indexed), vol. 4, no. 6, pp: 215-223, 2014.
4. M. Shoaib, A. Daud, M.S.H. Khiyal, "Un-supervised clustering approach for author name disambiguation." *Knowledge-based Systems*, under revision.
5. M. Shoaib and A. Daud, "Author name disambiguation in bibliographic databases, a Survey." *Frontiers of Computer Science*, under review.

Table of Contents

Table of Figures

Table of Tables

Table of Abbreviations

Abbreviations	Descriptions
ADC	Agglomerative Double Clustering
ACM	Association for Computing Machinery
AID	Author Identification Number
AND	Author Name Disambiguation
DBComp	DataBase Comparison
BDs	Bibliographic Databases
CbTM	Constraint-based Topic Modeling
CDC	Conglomerative Double Clustering
DBDL	DataBase Design Language
DBLP	Digital Bibliography & Library Project
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DLs	Digital Libraries
EM	Expectation Maximization
FCM	Fragment Comparison Method
GER	Grouped Entity Resolution
GHOST	GrapHical framewOrk for name diSambiguaTion
HHC	Heuristic-based Hierarchal Clustering
HMRF	Hidden Markov Random Fields
HPCs	High Precision Clusters
HRCs	High Recall Clusters
IDF	Inverse Document Frequency
IdRF	Identity Resolution Framework
IMDb	Internet Movie Database
LDA	Latent Dirichlet Allocation
LDA-ER	Latent Dirichlet Allocation-Entity Resolution
LOAD	Labeling Oriented Author Disambiguation
LSH	Locality Sensitive Hashing
MCMC	Markov Chain Monte Carlo
MeSH	Medical Subject Heading
MGP	Multi-level Graph Partitioning
MGPM	Multi-level Graph Partitioning and Merging
MLNs	Markov Logic Networks
MNDF	Maximum Normalized Document Frequency
MSF	Modified Sigmoid Function
PFG	Pair-wise Factor Graph
PLSA	Probabilistic Latent Semantic Analysis
RDF	Resource Description Framework
RSAC	Related Semantic Association based Clustering
SAM	Semantic Association Merging

SAND	Self-trained Associative Name Disambiguation
SDK	Shoaib, Daud, Sikandar
SHC	Seed-based Hierarchical Clustering
SVM	Support Vector Machine
TF	Term Frequency
VSM	Vector Space Model
UAIs	Universal Author Identifiers
UAIs_Sys	Universal Author Identifiers System
URLs	Universal Resource Locators

Chapter 1. Introduction

Chapter 1

Introduction

1.1. Preliminary

A real world entity may have multiple names and, on the other hand, multiple entities may be represented by a single name. This scenario is referred as entity ambiguity or uncertainty. To resolve the problem of entity ambiguity is called entity resolution. Entity resolution has many alternate terms like entity disambiguation [1], instance unification [2] web appearance disambiguation [3]. Entity ambiguity exists in human names because in every society of the world people share attractive common names. In the United States 300 most common male names are shared by more than 114 million people [4]. On the other hand, a person may be known by multiple names. In digital libraries (DLs) and bibliographic databases (BDs) it has been observed that multiple authors share a common name or a single author may appear with different names. This sharing or variation causes author name ambiguity in DLs and BDs. Name ambiguity is the foremost issue of BDs [5] and it causes unfair attribution to researchers' work, and affects the quality of services in BDs and in similar systems [6]. Resolving author's name ambiguity in citations is referred as *author name disambiguation* (AND).

A *bibliographic database* is an organized digital store of metadata of research publications, patents, books, and news articles, etc. Examples of bibliographic databases are: DBLP [7], CiteSeer [8], MEDLINE[5] and Google Scholar[6]. The metadata schema differs from one database to the other. Four types of publication features (co-authors, publication title, venue and publication year) are available almost in all BDs. Some BDs may store many other features, and there is no restriction to the number of attributes to be managed by a BD. These features are also called attributes, and we use these terms

[5] www.ncbi.nlm.nih.gov
[6] scholar.google.com

interchangeably. In this thesis we use AND specifically for author name disambiguation in bibliographic databases.

1.2. Author Name Disambiguation Problems

In literature many terms are used for this problem like name disambiguation [9] [10], object distinction [11], mixed and split citation [12], author disambiguation [13] and entity resolution [14] [15]. Name disambiguation problems can be divided into following three categories.

1.2.1. Name Synonymy/ Name Variant Problem

The problem of synonymy arises when an author has variations in his/her name in his/her citations. For example, the author name "Malik Sikandar Hayat Khiyal" is also written as "Sikandar Hayat" in citations of his publications. The DBLP treats them two different authors and divides his publications between two names (Table 1). In literature, this problem is also referred as name variant problem [16] [14], entity resolution problem [15], split citation problem [12] and aliasing problem [17].

Table 1: Example of name variant problem

Malik Sikandar Hayat Khiyal
Malik Sikandar Hayat Khiyal, Aihab Khan, SehrishAmjad, M. Shahid Khalil: Evaluating Effectiveness of Tamper Proofing on Dynamic Graph Software Watermarks CoRR abs/1001.1974: (2010)
Farhan Hassan Khan, Saba Bashir, M. Younus Javed, Aihab Khan, **Malik Sikandar Hayat Khiyal**: QoS Based Dynamic Web Services Composition & Execution CoRR abs/1003.1502: (2010)
Saba Bashir, Farhan Hassan Khan, M. YounusJaved, Aihab Khan, **Malik Sikandar Hayat Khiyal**: Indexer Based Dynamic Web Services Discovery CoRR abs/1003.1504: (2010)
Sikandar Hayat
Muhammad Imran Shafi, Muhammad Akram, **Sikandar Hayat**, Imran Sohail: Effectiveness of Intrusion Prevention Systems (IPS) in Fast Networks CoRR abs/1006.4546: (2010)
Muhammad Akram, Imran Sohail, **Sikandar Hayat**, Muhammad Imran Shafi, UmerSaeed: Search Engine Optimization Techniques Practiced in Organizations: A Study of Four Organizations CoRR abs/1006.4558: (2010)

1.2.2. Name Polysemy/ Name Sharing Problem

The problem of polysemy arises when multiple authors share the same name label or there exist name homonyms [18] in multiple citations. For example, "Guilin Chen" and "Guangyu Chen" write their names as "G. Chen" in their citations. It is quite possible that

a full name of an author is shared by multiple authors. Bibliographic databases may treat these different authors as a single author. Resultantly, on querying the database for such ambiguous names, it may list all citations under the single person's name. On querying DBLP against author name "Michael Johnson" it lists 32 citations and these citations are from five different people [14]. Table 2 shows first four citations listed by DBLP under the author name "Michael Johnson". In literature there are various names of this problem such as name disambiguation [9] [10] [19]), object distinction [11], mixed citations [12], author disambiguation [13] and the common name problem [14].

Table 2: First four citations (out of 32) listed by DBLP under author name "Michael Johnson"

Jennifer Mankoff, Susan R. Fussell, TawannaDillahunt, Rachel Glaves, Catherine Grevet, **Michael Johnson**, Deanna Matthews, H. Scott Matthews, Robert McGuire, Robert Thompson, Aubrey Shick, Leslie D. Setlock: StepGreen.org: Increasing Energy Saving Behaviors via Social Networks. ICWSM 2010
Michael Johnson: Barriers to innovation adoption: a study of e-markets. Industrial Management and Data Systems 110(2): 157-174 (2010)
Michael Johnson, Robert D. Rosebrugh, Richard Wood: Algebras and Update Strategies. J. UCS 16(5): 729-748 (2010)
Michael Johnson: Barriers to innovation adoption: a study of e-markets. Industrial Management and Data Systems 110(2): 157-174 (2010)

1.2.3. Name Mixing Problem

Shu et al [14] introduce another type of name disambiguation problem and refer it as name mixing problem. If multiple persons share multiple names it is called the name mixing problem. The two problems discussed above may occur simultaneously and cause the name mixing problem.

Typographical mistakes also cause name ambiguity problems. Treeratpituk and Giles [17] consider the typographical mistakes in names as a separate name disambiguation problem. These problems may arise due to use of abbreviations, spelling mistakes; and using caste or family name at the end or at the beginning of names. L. Branting [20] discusses nine different types of name variations.

1.3. Need for Name Disambiguation

Name ambiguity may distress document retrieval, affect web search and database integration. It may cause incorrect authorship identification in literary works resulting in

improper credit attribution to the authors. In academic digital libraries, disambiguating author names is necessary for following reasons:

- Users are interested to find publications of a particular author.
- It helps in expert finding. Publishers can easily find reviewers.
- Research communities and institutions can track the achievements of their scholars.
- Academic promotion and grant funding requires measuring research work of a particular candidate. Funding organizations are interested in the output of the research for which they had funded.
- Uniquely identified and disambiguated author names are used in many tasks such as searching homepage and finding topics, as a particular author is interested in a single or few topics.

1.4. Hurdles to Resolve the Problem

Author disambiguation is not an easy task due to various hurdles and constraints present in the bibliographic databases [21]. Here we highlight those hurdles:

- *Lack of identifying information*: The identifier metadata is either incomplete or not available at all.
- *Multi-directional problem:* Multi-disciplinary articles authored by multiple persons from multiple institutions (nationwide or world-wide) may cause ‘multiple entity disambiguation’ problem.
- *Less number of publications by most of the authors*: The machine learning techniques used for AND give better results when a reasonable number of examples is available. This is only possible when the individual authors have produced many papers. In MEDLINE almost 46% of the authors have written only one publication [21]. The authors having one to few papers are problematic.
- *Heterogynous nature of bibliographic databases*: The bibliographic databases are heterogeneous in many ways, like: schema heterogeneity, discipline heterogeneity, language heterogeneity, attributes heterogeneity, etc.

- *Economic issue*: The construction of such a database that can accommodate and manage the world-wide researchers' community including all the disciplines, nations and languages is not only economically unfeasible but also probably impossible.

1.5. Current Issues

In this research thesis, we have focused the issues that have been least concentrated in literature. Un-supervised methods are affected by *selection of seed* [22], *scarcity of information* [23] [24] sources, inappropriate[7] *similarity functions* [24] and poor clustering algorithms [6]. Selection of cluster seed, especially in hierarchal clustering, has not been focused properly. Most of AND approaches exploit only title, co-authors and venue attributes of publications [6]. Co-authors attribute is the most informative source of information in AND process [25] [24]. Title of a publication has only a few words to represent the topic and the contents of the publication. Similarity venue has a few words to represent the research areas of all its publications. Few words cannot represent the topic of a publication or research areas of a journal or conference. We name this issue as *scarcity of information* problem. Use of existing similarity functions may not prove good performance. All un-supervised methods use existing similarity measures. AND approaches exploit classification or clustering algorithms previously designed for different problems. However, some works like Cota et al [25] and Ferreira et al [22] propose their own clustering algorithms to resolve AND problems but the problem is still unresolved [4] [24]. Most of the previous approaches require prior knowledge about the number of actual authors belonging to an ambiguous name, whereas in real scenarios this is hard to have this information. Above mentioned issues are summarized in following points:

- Exploiting inappropriate similarity measures for publication attributes
- References (reliable source of information that can reduce the scarcity of information problem) have not been utilized
- No focus for selecting seed of a cluster

[7] The term '*appropriate similarity function*' we mean a similarity measure that better fits the attribute and the scenario than other measures.

- Little efforts for finding automatically the number of actual authors sharing an ambiguous name

Estimating name and co-authors similarities have not been treated properly. In most of the research works they have been treated like estimating title and venue similarities using different similarity functions. Most of these functions compare tokens of two names blindly. A part of the name (say, first name) may have two or more different tokens. For example "Muhammad" (in Muhammad Shoaib) may also be written as "Mohd." or "M." Comparing name tokens blindly will consider all these tokens different from each other whereas they all represent the same token.

Scarcity of information may cause estimation of inappropriate[8] similarity value between two citations. References of publications may help improve the similarity among publications. They have not been utilized in AND problems for calculating similarity among publications.

Another issue faced by AND is prior knowledge about the number of actual authors. Unfortunately, this is very hard to get this information, and it is not available at all for new datasets in real scenarios.

Initial entries of clusters, especially the seed, play important role in clustering process. Initial few wrong entries, especially in hierarchical clustering, may affect performance adversely. That is why we focus on the selection of seed and successive initial entries too.

1.6. Our Contribution

We propose similarity measures for each type of attributes, a clustering algorithm and exploiting references as additional source of information. Our contribution is briefed as under:

We propose improved similarity measures for each type of publication attributes. We divide name tokens into five different categories and devise a similarity function that well accommodates them. Our proposed similarity function for co-authors attribute assigns

[8] The term 'appropriate similarity value' we mean a similarity value that is closer to the actual similarity value. Actual similarity value means the similarity value measured by comparing the complete scripts of two documents rather than comparing their titles or venues or abstracts, etc.

higher similarity value to a pair of citations if both citations share more common co-authors irrespective of the total number of co-authors. For title and venue attributes we propose a conditional absolute measure (CAM) and, for titles of references we propose SDK[9] index.

For grouping the publications of the same author, we propose seed-based hierarchical clustering (SHC) algorithm. The citation (publication) which has maximum accumulative co-authors similarity is selected as the seed.

We also propose to use references of publications as additional sources of information to overcome scarcity of information problem.

1.7. Concepts

Publication: *It refers to any published literary work like research paper, book chapter and report.* We use "*publication"* and "*academic document*" interchangeably.

Citation: *The complete reference to a publication in a bibliographic database. It usually contains names of co-authors, title, venue and year of publication, etc.*

Triplet Attributes

In this work, *title, co-authors* and *venue* attributes are reffered as *triplet attributes* or simply *triplets.*

References: *The bibliographic list given at the end of a publication.*

Ref-titles: *We combine all titles of references of a publication into one and name it as references titles or ref-titles for short.*

Ref-coauthors: *We combine all co-authors of references of a publication into one and name it as references co-authors or ref-coauthors for short.*

Document: *The word document has no specific meanings in this work. It means any text document. We, at some occasions, use this term to generalize the discussion. So document may mean a citation or a publication or any text document or even a text string.*

[9] Last names of authors (Shoaib, Daud, Khiyal) who proposed this index

Note: When we use "co-authors", "title" and "venue" they always mean (in this work) citation co-authors, citation title and citation venue because they are part of the citations. These attributes are called as citation attributes. "Ref-titles" and "ref-coauthors" are not present in citations instead they are part of publications. So we call them publication attributes or references attributes. All citation attributes are also publication attributes, but vice versa is not true. In short, we use attribute names as co-authors, title, venue and ref-titles. At some occasions we use citation and publication alternatively.

Absolute and Relative Similarity Value: *The similarity value that is equivalent to the proportion of common data to the total amount of data in two documents is referred as absolute similarity value or simply absolute value. For example, if two documents share 80% data, the value 0.8 (normalized between 0 and 1) is the absolute similarity value. The similarity value that is either greater or lesser than absolute value is considered as relative similarity value or simply relative value.*

Absolute Similarity Measure*: The similarity function that outputs absolute similarity values for two documents is referred as absolute similarity measure or simply absolute measure or absolute function.*

Relative Similarity Measure: *The similarity function that outputs relative similarity values for two documents is referred as relative similarity measure or simply a relative measure or relative function.*

Information Sources*:* We use attributes, features, evidences and information sources interchangeably.

Notations

Few notations that are used throughout this work are listed in Table 3. Other notations are listed in separate tables at appropriate places.

Table 3: Notations used throughout this work

Symbols	Sets	Description
A	$A= \{a_1, a_2, \ldots, a_k\}$ where a_i is the *ith* author; k is # of unique authors sharing an ambiguous name	Set of authors/persons sharing an ambiguous name
P	$P= \{p_1, p_2, \ldots, p_z\}$	Set of publications associated to an

	z is number of publications belonging to A	ambiguous name
C	$C = \{c_1, c_2, \ldots, c_z\}$ z is number of citations belonging to A	Set of citations associated to an ambiguous name
Note: P and C are different only in a sense that former denotes the complete publication whereas the later denotes citation of the publication. Number of publications is always equal to number of citations. That is why we reserve the same letter (z) to represent their number.		
D	$D = \{d_1, d_2, \ldots d_n\}$ where n is number of documents.	Collection of all documents
$\check{V}$	Set of unique words or vocabulary	All unique words in documents to be compared.
Sim		Similarity
$\check{T}$		Token (e.g., part of name)
T		Title of publication
T		Term or word
$\check{N}$		Name (co-author name)

1.8. Thesis Organization

The rest of the thesis is organized as: chapter 2 describes related work; chapter 3 explains improved similarity measures and their comparisons with existing similarity measures; chapter 4 explains the role of references in similarity estimation of publications; chapter 5 explains our AND approach; and chapter 6 gives summary and future directions. Next to chapter 6 we provide a list of research works referred in this thesis. At the end we show screen shorts of our AND application implemented in C#.Net.

Chapter 2. Related Work

Chapter 2

Related Work

In chapter 1 we have introduced AND problems and related concepts. Chapter 1 also describes the need for and hurdles in disambiguating authors in BDs or DLs. In this chapter, we describe the related research work in detail by categorizing previous works in different categories as classified by Shoaib et al[10]. Here we review AND approaches, similarity measures and references as information sources.

2.1. AND Approaches

AND approaches provide a variety of solutions [26] ranging from manual assignment by librarians [27] to unsupervised learning. This work can be categorized in many ways, such as machine learning and non-machine learning approaches, supervised and unsupervised learning approaches, probabilistic and non-probabilistic techniques, etc. Most of the researchers categorize AND approaches under supervised, unsupervised and semi-supervised learning [26]. Statistical relational learning, graph-based and Ontology-based approaches have also been applied. Most of the research works under these approaches fall under machine learning categories, but some works also fall under non-machine learning approaches [28]. The AND works may combine two or three categories to take advantages of multiple approaches and to achieve better performance. Graph-based approaches can be combined to any other approach. Statistical relational learning usually combines graph-based approaches. We categorize AND research works in machine learning, non-machine learning, graph-based and Ontology-based approaches. Graph-based works may exploit machine learning or non machine learning approaches, but we explain them under a separate heading.

[10] M. Shoaib and A. Daud, "Author Name Disambiguation in Bibliographic Databases, A Survey." *Knowledge Engineering Review*, under revision.

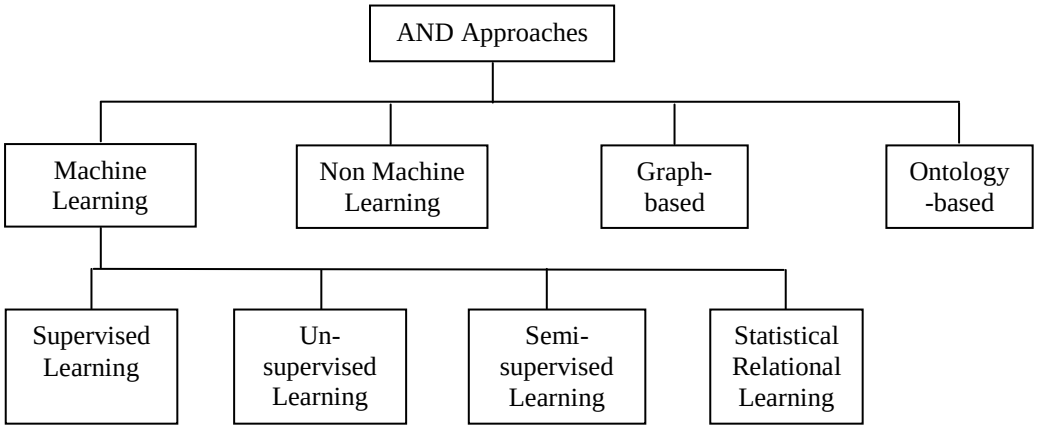

Figure 1: Hierarchy of AND Approaches

2.1.1. Machine Learning Approaches

We elaborate machine learning approaches under four sub headings: supervised, un-supervised, semi- supervised and statistical relational learning.

2.1.1.1. Supervised Learning Approaches

In this approach the data (citations) are divided into two parts: first, *training data* and second *test data*. Training data should be large enough for the classifier to extrapolate unseen data accurately. Almost 50% citations[11] of each author are used as training data and remaining as test data [29]. Training citations are labeled representing the corresponding author of the citations. The classifier models, say Naïve Bayes or SVM, etc. then learn through labeled examples (citations). Test data are used to predict unknown author of citations. Accuracy of model depends upon training on the data. This approach can predict the authors for whom we have trained the model. In other words, it cannot predict new authors. Skilled human annotators are required for labeling the citations for training. This makes learning process human dependent. Human errors can cause incorrect predictions for test data. Supervised learning is label intensive and requires much tedious work. Much care is needed while labeling with specific domain knowledge. This makes supervised learning label intensive, and error-prone if labeling or training of the dataset is not performed properly. Probabilistic models are popular to implement this approach.

[11] There is no hard and fast rule for proportion of the training data. It is normally 40%-60% of the total data.

In supervised learning [29] [30] [31] [17] [21] [32] [33] [34] [35] the major objective is to find labels in test data, say name labels by exploiting related rules learned in training phase. Supervised learning approaches achieve better performance and precision as compared to other methods [36] with the trade off expensive training time consumption, i.e., labeling labor [18]. Supervised approaches may exploit to predict an author name in a citation [29] or to disambiguate citations of a particular author [17] [31] [21] [30]. In following paragraphs we discuss supervised learning approaches by explaining different research works.

Han et al [29] proposed two supervised approaches to disambiguate author manes in the citations using VSM [37] for representation of citations; and cosine similarity for calculating the pair-wise similarity of citation attributes. They proposed canonical names by grouping together author names with the same first name initial and the same last name. Each canonical name is associated with all those citations where that name appears. First approach exploits a naive Bayes probability model [38] and the second support vector machines (SVMs) [39]. Both approaches exploit triplet attributes for similarity calculations. The difference between naïve Bayes and SVM is that the former model requires only positive examples to learn whereas the latter needs both positive and negative examples to learn how to classify citations. This famous work is actually an enhancement of Han et al [40] where they exploit k-means clustering along with a naïve Bayes model using same dataset and attribute set.

Torvik et al [30] proposed authority control, a framework for finding probability that the two citations sharing same author name belong to the same individual author. This framework resolves only name sharing problem for MEDLINE records. They use eight different attributes: (1) middle initial, (2) suffix (e.g., Prof. or II), (3) full name, (4) language, (5) number of common co-authors, (6) number of common title words, (7) number of common affiliation words and (8) number of common MeSH[12] words. They calculate the pair-wise similarity profile on the basis of these attributes and decide whether a pair of publications containing the same name of an author belongs to a single individual.

[12] Medical Subject Heading

Torvik and Smalheiser [21] enhanced the work of Torvik et al [30] by (a) including first name and its variants, emails, and correlations between last names and affiliation words; (b) employing new procedures of constructing huge training sets; (c) exploiting methods for calculating prior probability; (d) correcting transitivity violations by a weighted least squares algorithm; and (e) using agglomerative algorithm based on maximum likelihood for calculating clusters of articles that represent authors.

Culotta et al [31] proposed a model that overcomes the problem of transitivity produced due to pair-wise comparisons. A researcher can have multiple papers, email addresses and affiliations. While comparing publications of such authors the pair-wise classifier cannot handle multiple instances of an attribute. They employed sets rather than pair-wise comparisons, and addressed transitivity issue between co-authors in a better way. The comparison of a new citation is made with all citations of a cluster rather than pair-wise comparisons. By comparing a citation with sets makes it possible to handle multiple values of an attribute. They introduced cluster wise scoring function through error driven training and ranking based updating of parameters. They employ a greedy agglomerative algorithm for clustering purpose.

Yin et al [11] focused name sharing problem by considering only identical names. They proposed DISTINCT, an object distinction methodology to disambiguate authors. They combine set resemblance of neighbor tuples and random walk probability between two records of relational database. These two methods are complementary: one exploits neighborhood information of two records, and other uses connection strength of linkages by assigning weights. They apply SVM [39] to assign weights to various types of links in the graph and agglomerative hierarchical clustering to get final clusters.

Wang et al [41] proposed a two step model for name disambiguation that resolved only name sharing problem only for identical names in Arnetminer[13]. They proposed atomic clusters, i.e., each cluster had the citations of a particular author. At first step, they use a bias classifier to find atomic clusters. They used a list of publications having an ambiguous author name and triplet attributes of citations as input to the classifier. At the

[13] http://arnetminer.org

second step, they integrate atomic clustering[14] results into hierarchical and k-means clustering algorithms. Using atomic clustering they improved performance of disambiguation in terms of f-measure about 8% by hierarchical clustering method and 27% by k-means clustering method.

Treeratpituk and Giles [17] resolved name sharing problem in MEDLINE records. They introduced random forest classifier, a machine learning approach and find high-quality pair-wise linkage function. They define the similarity profile by considering 21 attributes categorizing them in six types of attributes; three of them are triplets and other three are: affiliation similarity, concept similarity and author similarity. They used a naive based blocking procedure. This procedure uses the author's last name and the first initial to block author name that does not share both parts of the author's name. They compared the results with SVM approach. Their results show that random forests outperform SVM by more than 2% on the average.

Wang et al [42] proposed a constraint-based topic modeling (CbTM) approach. Their work is actually the extension of Zhang et al [43]. They assume that if a pair of publications satisfies a constraint, then both the publications should have more chances to have similar topic distribution. They combined original likelihood function of latent Dirichlet allocation (LDA) [44] with a set of constraints defined over the attributes available from citations dataset. Thus likelihood function is also affected by constraints. In likelihood function, they used balancing factor restricting its value from 0 to 1. They define constraints as set of constraint functions each having value either 0 or 1. The function has the value 1 if a constraint is present in a pair of citations, and it is 0 otherwise. They define five constraints; two of them belong to triplet attributes excluding the title attribute and other three are: indirect co-author or transitive co-author (it is actually the τ-CoAuthor constraint defined in [43]); web constraint (it means that two publications appear in the same web page) and user feedback (what the users comment about two publications' authors). They use Gibbs sampling [45] for estimating model

[14] Wang et al [41] define it as "the atomic cluster means that publications in the same atomic cluster must be correctly grouped (high precision) but might be further grouped in the process of clustering (possible low recall)".

parameters and at the end agglomerative hierarchical clustering algorithm is employed to construct clusters for uniquely identified authors containing all their publications.

Qian et al [32] proposed LOAD (Labeling Oriented Author Disambiguation) to resolve author name disambiguation problem together with users. LOAD exploits supervised training for estimating similarity between publications using high precision clusters (HPCs) for each author to change labeling granularity from individual publications to clusters. Labeling HPCs decreases labeling effort at least 10 times as compared to the labeling publications. They found HPCs are clustered into high recall clusters (HRCs) to place all publications of one author into the same cluster. For pair-wise comparisons LOAD employs rich features like name, email, affiliation, homepage between two authors, co-author names, co-author emails, co-author affiliations, co-author homepages, title bigram, reference and download link. Besides, self citation and publishing year interval between two publications are also considered.

The approaches discussed above perform name disambiguation in an offline environment. Different from the above approaches, Sun et al [46] proposed citation analysis system. The focus of their approach is to decide, at query time by involving the user, if the queried author name matches the given set of publications retrieved from Google Scholar database. The system exploits two kinds of heuristic features: (1) number of citations per name variation, and (2) publication topic consistency. Topic consistency exploits discipline tags crowd-sourced from users of the Scholarometer system [47]. They train a binary classifier on a dataset of 500 top ranked authors from scholarometer database[15] by manually labeling either ambiguous or unambiguous, and examine the publications retrieved from Google Scholar for each queried name. To the best of our knowledge this is the first work addressing real-time author name disambiguation, and achieves 75% accuracy.

Although supervised machine learning techniques are the most effective methods among all other approaches yet they bear some drawbacks. One drawback of these methods is their scalability. They cannot classify correctly the author of a citation for whom training example(s) are not available. It is also infeasible to train thousands of models for all

[15]scholarometer.indiana.edu

individual authors in a large BD. Another drawback of these methods is that training data need to be large enough for classifiers to predict unseen data correctly. This introduces manual disambiguation of the large number of citations. Moreover, dynamic nature of BDs requires periodical training of the data for each individual author to capture new patterns.

The disambiguation process should be cost-effective but manual labeling causes heavy costs [48]. In BDs some authors may appear in several citations while the majority of the authors appear only a few times. Thus number of citations for an individual author may be extremely skewed [48] . The disambiguation task of less popular authors is challenging because only few examples are available for training the model.

These approaches are feasible when we need high accuracy, and we can bear labeling cost and wait for the manual labeling process to complete. These approaches are not suitable for citations of new authors for whom training examples are not available.

2.1.1.2. Unsupervised Learning Approaches

Unsupervised learning approaches [9] [10] [3] [49] [15] [50] [51] [52] [53] [54] [55] [56] [57] [35] [58] [59] [60] [61] need not manual labeling. Instead, they choose features to classify similar entities into clusters. Various clustering algorithms are applied to cluster similar entities. These methods automatically train data. Clustering algorithm is trained through unsupervised manner, i.e., there is no human interaction or labeling process. Parameter estimation plays key role in these approaches. The success of an algorithm depends upon similarity functions and parameter estimation. Markov Chain Monte Carlo (MCMC) [62], expectation maximization (EM) [63] and Gibbs sampling are commonly used parameter estimation tools. Unsupervised learning methods save labeling time with the trade off efficiency and precision. No doubt supervised approaches give high precision and recall, however in dynamic scenarios unsupervised learning methods are better solutions than supervised learning methods.

Unsupervised approaches may utilize similarities between citations with the help of predefined set of similarity functions to group the citations associated with a particular author. These functions are usually defined over the features present in citations [9] [10]

[52] [50] [53] [54]. These features are also called local information [14] as they are apparently available in citations. Similarity functions may also be defined over implicit information such as topics of the citation [14] [51] [55] or Web data [56] [57] [55]. Information about topic(s) of citation is not present in citation attributes, rather it is derived from corpus hence called global information [14]. Unsupervised techniques may also apply an iterative process to disambiguate authors of citations [15] [49]. In following paragraphs we discuss un-supervised learning approaches by explaining different research works.

Han et al [9] [49] improve their previous work [29] by applying k-way spectral clustering [9] and hierarchical naive Bayes mixture model [49] using triplet attributes as information sources. Malin [10] investigates two unsupervised approaches, hierarchical clustering and random walk through resolving name sharing and name variant problems.

Bekkerman and McCallum [3] resolved the name ambiguity problem. They presented two frameworks: first one used the link structure of the Web pages, and second exploited A/CDC (Agglomerative/ Conglomerative Double Clustering). Their approaches require a minimum of the prior knowledge as provided in bibliographic databases. However, their approaches best fit to web appearances instead of bibliographic databases.

Bhattacharya and Getoor [15] referred name disambiguation problem as entity resolution. They extended the LDA topic model [44] along with Gibbs sampling [45] for author name disambiguation. They suppose that the authors who belong to one or more groups of authors may co-author publications. They discover clusters of authors and clusters of publications written by these authors, simultaneously. They exploit an unsupervised approach to train the algorithm, and perform parameter estimation through expectation maximization (EM) algorithm along with Gibbs sampling [45]. Their model is about 100 times slower than an alternative approach [52], and solves only the name variant problem.

Bhattacharya and Getoor [52] proposed a collective entity resolution. This method is actually an improvement to their previous approach [15]. In this approach, they first assign publications to one individual author to assign them to other authors. Given two publications, both written by authors a_1 and a_2, if the two instances of a_2 refer to the same

individual, then it is likely that both instances of a_1 refer to the same entity. Resolving this 2nd level ambiguity helps in cases where there is a high level of ambiguity. They treat high verses low ambiguity scenarios separately. They first address the most confident assignments and then less confident ones. The final similarity value between the two citations is calculated on the basis of pair-wise comparisons and previously disambiguated authors. In other words, it is the weighted combination of feature similarity and relational similarity. The weighting parameter is adjusted manually and it may take different optimal values across different contexts. Although this model is advancement to their previous work Bhattacharya and Getoor [15] yet scalability is still a problem.

Song et al [51] proposed an algorithm based on probabilistic latent semantic analysis (PLSA) [64] and LDA [44] to resolved author name disambiguation exploiting the contents of articles. They exploited citation attributes, and publication's first page to relate authors to topics.

Shin et al [65] proposed name disambiguation framework constructing social network for finding semantic relationships between authors and solved name sharing and name variant problems simultaneously. They employ two methods; one for namesakes (name sharing) names and the other for heteronymous (name variant). The social network is constructed in three steps. 1) *Information extraction:* extraction of paper title, etc. 2) *Candidate topics extraction:* extraction of topics that are representative of the publication. These candidate topics are extracted from the abstract of the publication using morphemic analysis [66]. 3) *Social network construction*: the social network is constructed on the basis of above two types of information. They use cosine similarity metric for finding similarity among two social networks.

Yang et al [67] resolved name sharing problem by exploiting triplet attributes along with web attribute. They used cosine and *modified sigmoid function* (MSF) for triplet attributes, and *maximum normalized document frequency* (MNDF) for web attribute to estimate pair-wise similarity between citations. They also employed a binary classifier to reduce noise in clustering step.

Unsupervised learning approaches need not manually labeled training data [68]. They explore simply feature spaces, and often have lower performance than supervised learning techniques [69]. They save labeling efforts with the trade of performance (accuracy and efficiency). They are more suitable in a dynamic environment such as DLs and BDs than supervised approaches with a little loss of accuracy.

2.1.1.3. Semi-Supervised Learning Approaches

Semi-supervised learning approaches [70] [12] [71] [72] [14] [43] [22] [73] have also been applied to AND problems. It combines the characteristics of both approaches discussed above. In supervised learning, the labeling process causes additional cost to disambiguation task. However, annotating at least some examples improves disambiguation effectiveness. On the other hand, the acquisition of unlabeled citations is relatively inexpensive, but performance is not competitive to the supervised environment. The disambiguation process should be cost-effective and achieving high effectiveness. For achieving the advantages of both approaches semi-supervised techniques are exploited.

Shu et al [14] proposed LDA-dual topic model for complete entity resolution. They resolved name sharing, name variant and name mixing disambiguation problems simultaneously by extending LDA [44], a generative latent topic model, to LDA-dual. They introduced the concept of global information based on the words and author names present in the corpus. In LDA-dual they define topics as two Dirichlet distributions, one over words and the other over author names, characterizing topics as a series of words and author names. They also consider local information like publication title and co-authors, etc. along with triplet attributes they use topic similarity and minimum name distance. They claim that the two citations share little local information as compared to that of global information. They employed Metropolis-Hasting [74] within Gibbs sampling to calculate the global information, i.e., model hyper parameters: α, β and γ. They, based on these parameters, propose two algorithms and resolve all three problems simultaneously. Complete process consists of these steps: (1) finding topics of citations in the corpus using Gibbs sampling; (2) constructing a pair-wise classifier; (3) resolving name sharing problem with the help of spectral clustering and classifier's support for

each ambiguous author name; (4) solving name variant and name mixing problems with the help of the classifier.

On et al [70] proposed a framework for resolving name variant problem. They resolved AND problem in two steps: blocking step and distance measurement step. They used four blocking methods that reduce the candidates, and seven unsupervised distance measurements that measure the distance between the two candidate publications to decide whether they belonged to the same entity. They also exploited two supervised algorithms (naive Bayes probability model [38] and SVMs [39]) to separate the publications of an author in a cluster.

Lee et al [12] used naive Bayes model and SVM (both supervised methods); and cosine, TFIDF [75], Jaccard, Jaro and JaroWinkler (unsupervised methods) to resolve name disambiguation problem. They call name sharing problem as mixed citation and name variant as a split citation problem.

On et al [72] again focused their attention to name variant problem and call it grouped-entity resolution (GER) problem. They proposed quasi-clique, a graphical partition based approach. Unlike previous text similarity approaches like string distance, TFIDF or vector-based cosine metric, etc., their approach investigates the hidden relationship under grouped-entities using the quasi-clique technique.

Huang et al [71] resolved both types of problems on a small dataset selected from CiteSeer. They employed an online support vector machine algorithm (LASVM) as supervised leaner of finding the distance metric of citation attributes by pair-wise comparisons. The supervised learner easily handles new publications with online learning. For clustering the citations of the authors they use an unsupervised DBSCAN algorithm [48] that constructs the clusters on multiple pair wise similarities. The DBSCAN also handles the transitivity problem. They used different similarity measures for different attributes, e.g., edit distance for URLs and emails, Jaccard similarity for affiliations and addresses, Soft-TFIDF [76] for author names.

Zhang et al [43] proposed a semi-supervised name disambiguation probabilistic model that based on six constraints. They consider following constraints: (1-3) triplet attributes constraints; (4) CoOrg, an organization of both authors; (5) citation, one publication cites

the other; (6) τ-CoAuthor, two of the co-authors (one for each publication) are not same but they appear in another publication as co-authors. They exploited hidden Markov random fields (HMRF) for name disambiguation problem. They used EM algorithm [63] to learn the model for distance measures for ambiguous authors. They applied their technique on arnetminer[13]. Their model combines six types of constraints with Euclidean distance, and facilitates the user to refine the results.

Ferreira et al [22] proposed SAND (Self-training Associative Name Disambiguation), a hybrid approach in two steps. In the first (unsupervised) step clusters of authorship records are formed utilizing persistent patterns in the co-authorship graph. In the second (supervised) step training is performed through a subset of clusters constructed in the first step deriving the disambiguation function. They performed their experiments on DBLP and BDBComp collections. They claim that, SAND outperforms unsupervised approaches by 27% on DBLP and 4% on DBComp.

Supervised learning is costly and time consuming, on the other hand un-supervised learning is not as effective as supervised learning. We can employ semi-supervised learning approaches to minimize the disadvantages of supervised and unsupervised learning, and maximize the advantages of the both.

2.1.1.4. Statistical Relational Learning Approaches

Many machine learning problems enclose statistical (uncertainty) and relational (complexity) features. The standard approach for handling uncertainty is probability, and for complexity it is first-order logic [77]. We need learning and performing inference in such representational languages that can capture probability and first order logic [78]. This is the focus of statistical relational learning [79]. MLNs (Markov Logic Networks) also known as Markov random fields were developed to subsume statistical relational models [78]. MLNs extend first-order logic by attaching weights to formulae and combining it with probabilistic graphical models [77]. Popular inference methods used in MLNs are MCMC [62], Gibbs sampling [45] and belief propagation [62]. For learning MLNs, generative, discriminative and structure based weight learners can be employed [77].

MLNs have been successfully employed to entity resolution problems [80] [78] [81]. Song and Rudniy [80] employ Markov random fields to disambiguate biological entities. Singla and Domingos [82] exploit MLNs, and Yu and Lam [81] employ the dynamic structure of MLNs to de-duplicate citations, a type of entity resolution that investigates whether two citations represent the same citation. This problem matches to ours, i.e., author name disambiguation. Culotta and McCallum [78] also employ MLNs to de-duplicate citations and to disambiguate authors in citation records. We, here, explain the only author entity to be disambiguated through MLNs and Markov random fields.

Author ambiguity is usually resolved by constructing a vector of attributes for each pair of citations and applying transitive closure. A learned classifier (such as naive Bayes, logistic regression, etc.) is employed to predict whether they match. Many approaches of first order logic assume uniqueness of names. Markov logic removes this assumption using the standard logical approach, and introduces the equality predicate and its axioms: equality, reflexive, symmetric and transitive [82].

Equality: (Equals(x, y) or $x = y$ for short) and its axioms:

Reflexivity: $\forall x \;\; x = x$.

Symmetry: $\forall x,y \;\; x = y \Rightarrow y = x$.

Transitivity: $\forall x,y,z \;\; x = y \wedge y = z \Rightarrow x = z$.

Predicate equivalence: For each binary predicate R:

$\forall x_1, x_2, y_1, y_2 \;\; x_1 = x_2 \wedge y_1 = y_2 \Rightarrow (R(x_1, y_1) \Leftrightarrow R(x_2, y_2))$.

The citation database structure can be represented by relations as: Co-author (citation, co-author), Title (citation, title), and Venue (citation, venue) relate citations to their attributes; HasWord (title/venue, word) and HasName (co-authors, name) indicate which words and names are present in title/venue attributes and in co-authors attribute; SameVenue (venue, venue) represents venue equivalence; and SameAuthor (author, author) predicts author equivalence. The truth values of all relations except for equivalence relations are provided in the citation database. The objective is to predict the SameAuthor relation, i.e., whether the ambiguous name in two citations is the same. The whole dataset can be represented through different types of predicates in MLNs. As an

example, a transitive predicate for venue attribute (given below) means: if venues of c_1 and c_2 are same and those of c_2 and c_3 are same then venues of c_1 and c_3 are also same.

SameVenue ($venue_1$, $vunue_2$) ∧ SameVenue ($vunue_2$, $vunue_3$)) SameVenue ($vunue_1$, $vunue_3$)

The weights of relations are learned and truth values of equal predicates are counted to participate in the prediction of author entity. It is usual that the attributes do not mach completely rather part of an attribute matches with the corresponding attribute of other citations. In such situation SameAttribute ($attr_1$, $attr_2$) gives truth value equal to 0 whereas the corresponding attributes may have many (if not all) words common. Here HasToken ($attr_1$, $attr_2$) may be exploited or alternatively cosine like similarity measures can be employed.

Tang et al [4] proposed unified probabilistic AND framework. They formalized the problem in hidden Markov random fields (HMRF [83] by exploiting relationships (like coPubVenue, citing[16], etc) and attributes (like title, venue, etc) in a two-step algorithm for parameter estimation. *CoPubVenue* relationship means if both the publications share the same venue; and *citing* means if a publication cites the other publication. They model publication data into an undirected informative graph, in which each node represents a publication and each edge a relationship. Attributes of a publication are associated to the corresponding node as a feature vector. Tang et al [4] define the publication informative graph as:

"Given a set of publications $P = \{p_1, p_2, \ldots p_z\}$, let $\check{r}_k(p_i, p_j)$ be a relationship between p_i and p_j. A publication informative graph is a graph $G = \{P; R; V_P; W_R\}$, where each $v(p_i) \in V_P$ corresponds to the feature vector of publication p_i and $w_k \in W_R$ denotes the weight of relationship $\check{r}_k$. Let $\check{r}_k(p_i, p_j)=1$ if there is a relationship $\check{r}_k$ between p_i and p_j; otherwise, $\check{r}_k(p_i, p_j) = 0$."

Tang et al [4] formalized the contribution of each attribute and each relationship as the weights of the feature functions. The objective function (Lmax) in the HMRF model is a

[16] Note: Tang et al [4] use the term "citation" but we have replaced it with "citing" to avoid ambiguity as we have defined citation in another way.

posterior probability distribution of hidden variables (authors) given observations (publications). Tang et al [4] formulize it as:

$$L_{max} = \log\left(\frac{1}{Z_1' Z_1'} exp\left(\sum_{(y_i,y_j)\in E,k} \lambda_k f_k(y_i, y_j) + \sum_{(x_i)\in X,l} \alpha_l f_l(y_i, x_i)\right)\right) \quad (1)$$

Where λ_k and α_l are sets of weights (parameters) for relationships and attributes respectively and are estimated while parameter estimation; $f_k(y_i, y_j)$ and $f_k(y_i, x_i)$ represent relationship (edge) feature functions and attribute (node) feature functions respectively; $Z_1' Z_1'$ is normalizing factor; E denotes the set of relationships; and X denotes the set of publications. Equation 1 clearly depicts the formulization of both types of publication data.

Tang et al [4] exploited a learning algorithm for parameter estimation that consists of two iterative steps: *assignment* of publications, and *update* of parameters. Initially the parameters are randomly initialized and publications are assigned to the candidate author clusters. The number of candidate/real author (clusters) may be provided as user input, but Tang et al [4] introduced an algorithm that automatically estimates the real number of authors. For initializing the cluster centroid they exploit graph partitioning method to identify atomic clusters (the clusters whose publications have similarity greater than the threshold). They greedily assign publications by selecting the publication that has the highest similarity to the cluster centroid. The publications which have similarity less than threshold are assigned to disjoint atomic clusters. After the initial step the centroid of each cluster is estimated and the weight of each feature function is updated maximizing the objective function. For parameter estimation, they employ contrastive divergence algorithm [84], which approximates the distribution by several Gibbs sampling [45] steps.

The statistical learning approaches support supervised, unsupervised and semi-supervised learning and provide better accuracy (Tang et al [4]) as they benefit from local contents as well as from relationships. On the other hand, they are computationally expensive. From the efficiency performance results given in Tang et al [4] we note that their

approach is 546% more expensive than x-means (the least expensive)[17], and 25% more expensive than hierarchal agglomerative clustering (the most expensive). These approaches are feasible only if the relational features of data are available.

2.1.2. Non Machine Learning Approaches

Most of the AND approaches utilize machine learning techniques however some works exploit non machine learning techniques too. They exploit heuristics and similarity based clustering techniques and do not need any training or learning [85] which may be quite expensive and time consuming too. For non learning approaches, instead of training and learning, we optimize parameters such as similarity thresholds. These approaches exploit similarity measures for available attributes and employ heuristics on the base of common pattern present in the dataset. The success of this approach totally depends upon the similarity functions and the heuristics. Research works such as [36] [86] [87] fall in this category. Such methods usually employ co-authors lists for initial clustering. For example two citations c_i and c_j can be combined to initialize a cluster corresponding to particular author if they share at least one co-author [87]. This cluster can be compared with the remaining citations. If the similarity between the cluster and the query citation, say c_{j+1}, exceeds a threshold value then the citation c_{j+1} is combined to the cluster else a new cluster is created. When all the citations are assigned to the clusters then other attributes like title, venue, etc can be employed to help combine the clusters. The two clusters may be combined (fused) if their inter cluster similarity is greater than threshold.

Oliveira et al [36] exploited fragment comparison method, a pattern matching algorithm, to cluster names that presented some degree of similarity to each other. The algorithm takes the strings of two corresponding author names represented in some canonical form as input, and compares them using an edit distance measure ignoring the order of the tokens that compose the names. If the distance between two author names is less than a threshold, then other attributes are used as additional evidence to determine whether the two names correspond to the same author and can be joined to the same cluster. This

[17] The least and the most expensive approaches we mean "out of those five approaches which are compared in reference [81]."

strategy tends to generate too many clusters when there is not enough evidence to disambiguate a name.

To overcome this issue, Cota et al [25] proposed a two-phase heuristic-based hierarchical clustering method. They determine the similarity between the names of authors by exploiting the same fragment comparison algorithm as Oliveira et al [36] do. In the first phase, they create clusters of citations having at least one compatible co-author on the general observation that only very rarely two ambiguous authors share a co-author. The resulting clusters are almost pure (with few wrong entries) but fragmented[18]. To decrease the fragmentation they use second phase where they combine clusters of citations of compatible author names based on several heuristics and similarity measures between the citation attributes. This phase is successively repeated until no more fusions are possible. They estimate the similarity among clusters considering the contents of titles and venues as bags of words instead of estimating their similarity by the summation of the similarities of their corresponding records (Malin [10]). Their technique is significantly less expensive than that of Malin's because they do not estimate the similarity among all pairs of entries in clusters, but only the similarity between the bags of words that represent them. Another superiority of this work is that it needs not to know the number of actual authors in advance as required by supervised approaches.

Carvalho et al [5] enhanced the work of Cota et al [25] and, on each load, assigned the new citations to their correct authors in already disambiguated database in an incremental way. For this they need not to disambiguate the whole database as it is required in a supervised learning environment. The method assigns new citations to the authors (clusters) with similar citation records or to new authors (clusters) when the similarity evidence is not strong enough. The method employs specific heuristics and similarity functions for ensuring whether new citations belong to pre-existing authors of the BD or if they belong to new ones without running the disambiguation process in the entire database.

Strotmann et al [87] disambiguated author names by exploiting deterministic clustering algorithm based on heuristics and well defined similarity measures without employing AI

[18] Citations related to the same author are spread in different clusters

techniques. They employ collaboration network in which author occurrences are presented by nodes and common evidences (attributes) by links between the nodes. The nodes are merged if two publications have at least one common co-author. The similarity of two citations is measured on the base of common co-authors, venues and common topics of the publications. The similarity of all the attributes is added and then nodes are merged if the similarity value between the two citations exceeds predefined threshold.

These approaches are not only easy to implement, but also save human efforts spent in training the data in supervised and semi-supervised learning approaches. These approaches are more feasible when we have limited computing resources and/or we need the disambiguation output as early as possible.

2.1.3. Graph-based Approaches

The graphical representation of citation datasets varies in the literature. Here we describe the graphical representation approach proposed by Fan et al in reference [19]. The publication dataset *P*, constructed from an ambiguous author name, can be represented as an undirected graph G = {V, E}, where each node $v_i \in V$ represents an ambiguous author name *A* or an instance of the queried author name "a_i" in a certain publication. Each edge ($(v_i, v_j) \in E$) between two nodes represents co-authorship relation between two authors. Each edge between v_i and v_j has also a label denoting the set of publications co-authored by the corresponding authors of v_i and v_j. In this way we collect all publications co-authored by v_i and v_j.

Suppose the following synthetic tiny dataset (example 1). In this example "A. Daud", an ambiguous author name, is present in both of the publications. We want to make sure if "A. Daud" in both publications is the same person or two different persons. The graphical representation of this example is shown in figure 2. Here we represent only co-authorship graph so we do not show the titles of the citations (publications).

Example 1

P_1: M. Shoaib, A. Daud, S. H. Khiyal, M. Sher. SIGMOD'02.

P_2: M. Shoaib, A. Rahim, M. Sher, A. Daud. CIKM'02.

In figure 2, two nodes are labeled as "A. Daud". Each such node represents an instance of "A. Daud". There is an edge between the nodes "M. Shoaib" and "A. Rahim" showing that they have co-authored publication p_2. Similarly the edge between the nodes "M. Shoaib" and "M. Sher" shows that they have co-authored publications p_1 and p_2. For simplicity, we ignore other types of relationships and attributes. If we manually analyze figure 2, it clearly depicts that the probability that "A. Daud" is the same person in both of the publications is very high, as a_1 and a_2 both are connected to the nodes "M. Shoaib" and "M. Sher" through p_1 and p_2 respectively. In other words, p_1 and p_2 have two common co-authors (M. Shoaib and A. Rahim) making it more probable that a_1 and a_2 represent the same person.

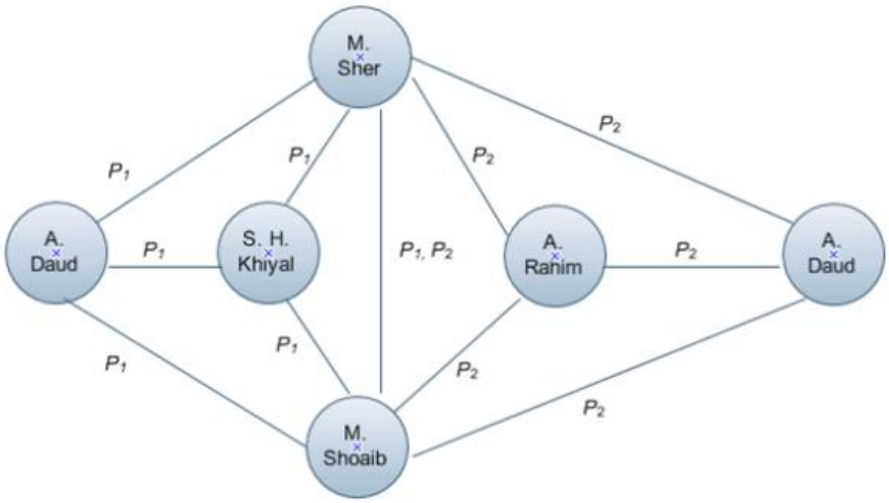

Figure 2: Co-authorship graph of the two citations

Figure 2 represents only the co-authorship graph. Any type of relationship (co-venue, co-organization, self-citation, etc.) can be represented through a graph.

The graph-based approaches are popular in name disambiguation problems. They can be employed in any of the AND approach. Many works employ co-authorship graph to capture the similarity between any two objects. It has been adopted by many approaches discussed above, such as *relational similarity* in Bhattacharya and Getoor [52] and Yin et al [11]; inter-object *connection strength* in Kalashnikov and Mehrotra [88], Yin et al [11] and Chen et al[89]; and *semantic association* in Jin et al[90]. The length of the shortest path in a graph is usually employed to estimate the degree of closeness between two nodes. Kalashnikov and Mehrotra [88] and Yin et al [11] utilize connection strengths to

find similarity of two nodes connected to each other through relationships. For this purpose Fan et al [19] exploit valid paths[19] and Kalashnikov and Mehrotra [88] name valid path as legal path. Bhattacharya and Getoor [52] employ collaboration paths of length three and assign equal weights to all paths regardless of their length. Kalashnikov and Mehrotra [88] propose a complicated method to calculate weights for connection strengths. They propose multiple equations and an iterative method to determine the weights. Differently from them, On et al [72] exploit *quasi-clique*, a graph mining technique [91] to take advantage of contextual similarity in addition to syntactic similarity. On et al [72], Chen et al [89] and Jin et al [90] estimate the similarity between two nodes (authors) as a combination of the feature-based similarity and the connection strength of the graph. Chen et al [89] estimate the connection strength between two nodes as the sum of connection strengths of all simple paths no longer than a user-defined length.

In the above paragraph we presented short but the comparative description of some of the graph-based works. In following paragraphs, to make the discussion more comprehensive, we describe some important works separately.

Jin et al [90] proposed SAND (Semantic Association Name Disambiguation), a graph-based approach. The similarity between the attributes (expect co-authors) of the two publications is measured through VSM, and TFIDF [75] is used for term weight. For co-authors and transitive co-authors semantic association graphs are constructed. The nodes show authors and the edges represent the association. The edges also determine the weight by counting the number of publications co-authored by two authors. SAND is a two step process, RSAC (Related Semantic Association based Clustering) and SAM (Semantic Association based Merging). RSAC clusters two publications in a group if the co-authorship graphs of the two publications are similar, i.e., they have common co-authors. Similarly, all the publications are grouped in small clusters. It is quite possible that the transitivity property holds true for co-authors of some publications but RSAC does not handle it, and all the publications of an author may be assigned to multiple groups. To handle this issue SAM merges groups on the basis of similarity values

[19] A path is valid if it contains no invalid vertex. A vertex is invalid if an intermediate vertex in a specific path and its two adjacent vertices form a triangle-like structure.

calculated for literatures (titles + abstracts), affiliations and transitive co-authorship graphs.

McRae-Spencer and Shadbolt [92] resolved the name disambiguation problem on large scale citation networks through graph-based approach exploiting self-citation, co-authorship and document source analyses in three passes to tie the publications of a particular author in a collection assigned to that author. The first pass is to test each publication in the ambiguous name cluster against every other publication within that cluster to see if the second publication is the self-citation of the first, or vice versa. Similarly the second pass is performed to draw a co-authorship graph, and the third pass uses source URL metadata. The output of these three passes is the graphical representation of the citations. This approach is based on metadata rather than textual context and on the notion that authors cite their own previous publications. As this approach uses self-citation as an attribute so the new publications have fewer or may have no citations at all. The publications of an author written just before his/her retirement[20] or death will never have self-citations. Similarly the publications written just before the change of research area will be self-cited hardly ever. This factor can decrease the accuracy of the approach.

Yin et al [11] proposed DISTINCT, an object distinction methodology and one of the state-of-the-art algorithms to solve name ambiguity problems where entities had identical names. Their method combines set resemblance of neighbor tuples and random walk probability (between two records in the graph of relational data) to measure relational similarity between the records of relational database. These two methods are complementary: one exploits the neighborhood information of the two records, and the other uses connection strength of linkages by assigning weights. DISTINCT exploits several types of linkages, like title, venue, publisher, year, authors' affiliation, etc. The method applies SVM on constructed training sets to determine weights to various types of linkages in the graph, and exploits agglomerative hierarchical clustering to get final clusters.

[20] By the term "retirement" we do not mean the retirement from job rather we mean retirement from research work willingly or unwillingly due to any reason.

Fan et al [19] resolved name sharing problem through GHOST (GrapHical framewOrk for name diSambiguTion) exploiting only co-authorship attribute, however for dense authors they exploited user feedback too. Contrary to the approaches of Chen et al [89] and Jin et al [90], GHOST does not take into account the feature-based similarity, and the connection strength between nodes v_i *and* v_j is measured using *Ohm's Law*-like formula defined over a subset of valid paths. Another difference of this work from the work in [90] is that it does not model the transitive co-authorship graph. This work has two beauties. *First*, the time complexity is very low as compared to the previous works as it exploits only co-authors attribute and achieves 94% precision on the average. *Second*, GHOST employs *Ohm's Law*-like formula to compute similarity between any pair of nodes in a co-authorship graph. The drawback of GHOST is that the results for dense authors are not in line with the results of non dense authors. Fan et al [19] proposed user feedback for such authors. No doubt the results are improved, but the scalability is a challenge over here because in real life databases there may be thousands of dense authors.

Wang et al [93] proposed active user name disambiguation (ADANA) exploiting a pair-wise factor graph (PFG) model which could automatically determine the number of distinct names. Based on PFG model, they introduced a disambiguation algorithm that improved performance through user interaction. Conceive

The works that utilize graphical approaches basically exploit the relationships among the publications by representing them in graph form. The graph structure is then analyzed to find the similarity between the two publications, along with other attributes. The remarkable question about graphical approach is that if it is possible to represent the titles and topics of the publications to estimate the corresponding similarities through the graph. For topic attribute, the answer "yes" seems to be almost impossible as the two publications will never share the same topics with the same probabilities, except the two publications are actually the copies of each other. For title attribute, the answer is "yes" at the words level, but it is not feasible to employ it as the two titles, even if by the same author, normally share a small proportion of words or may not share any word at all. The VSM or string matching approaches are better solutions for this attribute.

On et al [18] addressed the scalability of the name disambiguation problem, by exploiting two multilevel algorithms: the *Multi-level Graph Partitioning* (MGP) algorithm, and the *Multi-level Graph Partitioning and Merging* (MGPM) algorithm. Multi-level algorithms follow four major steps: 1) the input is represented as an affinity[21] graph; 2) the graph is divided into smaller graphs level by level; 3) the smallest graph is clustered at the end; and 4) partitioned graphs are restored to the size of the original graph. On et al [18] claimed that MGP algorithm provided better precision, but slightly lower recall than the spectral clustering methods. However, in terms of scalability, it outperformed by orders of magnitude up to 157 times faster than spectral clustering in DBLP. On the other hand MGPM improved performance with similar or better accuracy. For this purpose they employed MGPM algorithm, in which the merging step was included in the multi-level graph partitioning algorithm. By exploiting MGP and MGPM they achieved scalability and accuracy respectively.

Graph-based approaches can be combined with any type of AND approaches because they are actually the representation of the data and getting information from graph may suit to any type of AND approach.

2.1.4. Ontology-based Approaches

In information science, an ontology is basically the knowledge of concepts and the relationships between those concepts within a domain. In other words, it is a way of representing the domain knowledge.

Ontology-based name entity identification has been exploited by many researchers in different fields. For example, geographic named entity disambiguation [1], IdRF (Identity Resolution Framework) [94], named entity disambiguation exploiting Wikipedia [95] [96] entity co-reference [97], etc. So far as bibliographic databases are concerned researchers paid little attention to ontological approach. We could find only a few works in this particular domain. To the best of our knowledge reference [98] is the first published work in his field.

[21] It is similar to G = {V, E} as explained in example 1 (section: graph-based approaches).

Hassel et al [98] resolved author name disambiguation through already populated ontology extracted from the DBLP [7]. They utilize a file from DBLP that contains objects like authors, conferences and journals, and convert it into RDF (Resource Description Framework)[22],[23] and use it as background knowledge. Their approach takes a set of documents from DBWorld[24] posts, "call for papers" to disambiguate authors. Each such document contains multiple authors, say, the committee members, and some information about them, like affiliation; and information about the venue like topics of the venue. The scenario of the approach is different from those we have discussed throughout this article. All other approaches perform disambiguation by either predicting the most probable author of a citation or by grouping the citations by the same author in a unique cluster. Different from them, this approach pinpoints, with high accuracy, the correct author in the DBLP ontology file that a document (of DBWorld) refers to. Their method selects an author name from the document and searches the candidate authors in the populated ontology in RDF form. All candidate authors are compared with the author in the document to predict the most confident author in the ontology that relates to the author of the document. Different types of relationships are exploited to predict correct author out of various matches (candidates) in ontology. These relationships include entity name, text proximity, text co-occurrence, popular entities and semantic relationships. Name entity refers to specifying which entities from the populated ontology are to be spotted in the text of the document and later disambiguated as all the entities of the document may not present in the DBLP ontology. Text proximity is the number of space characters between the name entity and the known affiliation. Here known affiliation means the object already known by the ontology as affiliation, say, the name of a university. In DBWorld postings, affiliations are usually written next to the entity name. If an entity name in a document and an affiliation matches an author name and known affiliation in the ontology, there are chances that these two entities are the same real world entity. Text co-occurrence is utilized to match research areas of the candidate authors in the ontology and. Popular entity is the author in an ontology that has the

[22] World Wide Web Consortium: Resource Description Framework (RDF) Schema Specification 1.0. Retrieved on August 28, 2012 from http://www.w3.org/TR/2000/CR-rdf-schema-20000327/ (2000)
[23] World Wide Web Consortium: Survey of RDF/Triple Data Stores. Retrieved on August 30, 2012 fromhttp://www.w3.org/2001/05/rdf-ds/DataStore (2001)
[24]DBWorld. http://www.cs.wisc.edu/dbworld/ April 9, 2006.

highest score of publications among the candidate authors. Semantic relationships are used to match co-authors of candidate authors in an ontology, and entities in the document, with a notion that the entities in a document may be related to one another.

Park et al [99] proposed OnCu System to resolve name sharing problem through ontology-based category utility. Term category utility was used for similarity measurement between two entities. They exploit two types of ontology: *author ontology*, built on the publications from several proceedings of conferences, and the computer science *domain ontology*. Different from Hassel et al's scenario [98], they determine the correct author from various candidate authors in the author ontology exploiting the domain ontology for estimating the semantic similarity. Their goal is to discover the right author of the input publication and his/her right homepage. Their approach also differs from that of Hassel et al's [98] in using ontology-based evaluation functions. OnCU views candidate authors as clusters of their publications and employs a cluster-based evaluation function, exploiting ontology to predict the right author out of multiple candidate authors. Although OnCu's scenario also does not match exactly with that of we are surveying yet it is more related to ours as compared to the Hassel et al's [98]. Further, their approach can easily be fitted to our scenario.

The ontology-based approaches provide better semantic similarity measures for different attributes, but this is fruitful only if the ontologies providing background knowledge are frequently revised to meet the dynamic nature of the digital libraries.

2.1.5. Comparison of Proposed Approach with Baseline Methods

We have discussed a number of approaches in AND process. The nearest works to our approach are Cota et al [25] and Ferreira et al [22]. In spite of many common steps our work is different from their works.

Cota et al [25] and Ferreira et al [22] use fragment comparison method (FCM) for name matching and Jaccord coefficient for co-authors attribute in the first phase of clustering whereas we use our own proposed similarity measure for estimating name and co-author similarity. They start from very first citation as the seed of the cluster where as we select the citation as the seed which has maximum accumulative co-author similarity. Cota et al

[25] and Ferreira et al [22] in second phase combine the titles and venues of all the citations of the clusters produced in the first phase. Cota et al l [25] in second phase merge the clusters if the title or venue information of two clusters has similarity score greater than user provided threshold. The process continues until no more clusters are merged. Ferreira et al [22] in second phase use the clusters created in phase I as training data for non clustered citations and also for merging the clusters if the trained model favors to combine the citations of two clusters. In the second phase, they also use title and venue attributes as information sources.

In contrast to both methods described above we, in the second phase, arrange the clusters created in the first phase in descending order of size. Many clusters may have just one citation in first phase. We do not merge the information of attributes to compare the clusters for merging. We compare the smallest cluster with the seed of the biggest cluster and combine them if the similarity score of majority of attributes is greater than the threshold value. If the similarity score of majority of attributes is less than the threshold then the second largest cluster is invested for merging. This process continues for all clusters bigger than the smallest cluster. The same process is repeated for second smallest cluster, third smallest cluster, and so on. The whole process of the second phase is repeated until no more merging is possible. In case of two clusters are tested to be combined the seed of the biggest cluster is compared with the first citation of the smaller cluster. If the similarity score of majority of attributes is greater than the threshold then they are merged otherwise next citation of citation is compared. The two clusters are merged if any of the citation of smaller cluster has a similarity score of majority of attributes greater than threshold. Comparing two or fewer citations than comparing the whole clusters saves similarity calculation time with the same or better results.

2.2. Similarity Measures

Many works in document clustering like [100] [101] and in author name disambiguation such as [14] [15] [42] [51] use topical information [102] to group the similar documents. Donald et al [103] evaluate similarity measures that exploit topical like information present in documents. Rafi and Sheikh [100] propose a similarity measure based on the topic maps representation of documents. Wan [104] proposes document similarity

measure based on the earth mover's distance. These both works try to find subtopics similarity in documents. Our works (Shoaib et al [105] and Shoaib et al [106]) are different from them in a sense that we focus keyword matching.

In the literature, many similarity measures such as Jaccard coefficient, Manhattan, Euclidean, Pearson correlation, Kullback-Leibler divergence, chi-square, Dice and overlap are proposed for comparison of two documents. Out of similarity measures discussed in literature cosine is the most popular for measuring documents similarity [104] [107] [100] [108] [109]. In document clustering works like [104] [108] [101] and in author name disambiguation works like [12] [13] [29] [70] [43] [22] [25] exploit cosine measure representing the documents in VSM [37].

Pandit & Gupta [110] provide comparative study on distance measuring approaches and Cohen et al [111] and Branting [20] compares different similarity measures for name matching. Lee et al [112] provide empirical evaluation of models of text document similarity. Teghva and Veni [107] and Strehl et al [101] evaluate effects of similarity measures on document clustering, and Shoaib and Daud in their unpublished work[10] give brief overview of similarity measures used in AND. Arif et al prove that hybrid similarity measures instead of single similarity measure for all attributers is a better choice [60]. Strehl et al [101] use YAHOO datasets already categorized by human experts in different categories. In order to evaluate different similarity measures they perform several different clustering algorithms exploiting four different similarity measures (Euclidean, cosine, Pearson correlation and extended Jaccard). The experiments show that extended Jaccard and cosine measures are very close to human performed results [100]. All these studies are generic, and are conducted on existing measures. We could not find any work that compares different similarity measures for publications or proposes new similarity measures specially designed for them.

Selecting apposite similarity metric is a technical and challenging task [110] in Data Mining. It is advisable to employ the best fit similarity measure for each attribute of the citations. In literature, different types of similarity measures such as syntactic (VSM [37]), semantic (topic based like [29]), graph-based ([11] [16] [19] [93]) and ontological ([1] [97]), have been exploited.

In AND literature estimation of name similarity is not much focused. Most of the research works estimate names similarity of co-authors through existing similarity measures (Dice's coefficient in [14] or cosine in [29]). Cota et al [25] consider two names $ň_i$ and $ň_j$ as compatible if they have the same first initial and the last name, and identical if all the tokens of $ň_i$ exactly match to those of $ň_j$'s. They assign equal weight to identical and compatible names, whereas identical names should be assigned larger similarity value as compared to compatible ones. Shu et al [14] make the difference between identical and compatible names and assign distance 0 (similarity 1) to former case and 0.5 to the later one. Though this is more appropriate yet it does not count for the degree of similarity between $ň_i$ and $ň_j$. We, in Shoaib et al [105], differentiate between identical and compatible names, and also count for the degree of similarity between compatible names.

Ferreira et al [22] and Cota et al [25] standardize co-author names using only the initial letter of the first name along with the full last name. Standardizing this way may convert two different names to the same name. Suppose "Ajay Gupta" is a co-author of citation c_i and "Akaash Gupta" is a co-author of citation c_j, after normalizing they both become "A. Gupta" and cause false decision made on the base of Standardizing method. To avoid such scenarios Shoaib et al [105] estimate co-authors names similarity in a different style.

Shoaib et al [105] proposed four different similarity measures (for 4 different types of data) for academic publications. The first two measures, deal with name similarity and co-authors similarity respectively; third and fourth measures are designed for short and long documents. Third measure gives absolute similarity output provided that the documents do not repeat any term. Fourth measure, named as Shoaib index, tries to output absolute value.

Existing similarity functions are not absolute measures rather they are relative measures. Relative measures do not show the real picture of shared data between two documents. We, in Shoaib et al [106], propose SDK index that provides proportional weights to a number of common, non common, frequent and rare words assigning logarithmic weights to their frequencies. We are concerned to estimate syntactic similarity only, and not the semantic similarity. Syntactic similarity methods (such as cosine, Jaccard) are those

which compare two documents blindly and are unaware of the context and semantics of the word used. On the other hand, semantic similarity approaches such as topic modeling methods [44] [102] and WordNet based [113] approaches are aware of the meanings and context of the word used.

2.3. Publication Attributes

Publication attributes are the basic source of information and play important role in similarity estimation. Most of the works in author name disambiguation like Han et al [29] use *triplet* attributes for estimating similarity among publications. Almost half of AND works use only these three citations attributes [6]. Works in AND exploit diverse types of attributes such as self citation [4] [43], abstract [4] [43], user feedback [42] [19] topic of the publication [29] [42] [14] [51] [15], author affiliation [43], authors email addresses [43], web information [42]. Shu et al [14] use latent dirichlet allocation [44] for topic modeling [102]. Kleb and Volz [1] use ontological or semantic [113] techniques for guessing topics the publications. Torvik et al [30] use eight different attributes available in MEDLINE records. Smalheiser and Torvik [26] enhance their task of [30] by including first name and its variants, emails, and correlations between last names and affiliation words.

In the field of academic document clustering the use of reference markers and the context surrounding them has also gained much attention [114] [115]. In the works of Mercer and Marco [114] and Nanba et al [115], text surrounding a reference marker is extracted to determine the relatedness between the two publications connected by that reference marker. Aljaber et al [116] use contexts of reference markers to optimize similarity among publications. Another work by Jeon Hocheol [117] crawls the comments related to the papers cited in the related works sections and then provides useful information regarding the cited papers and how much similar are the cited papers and the paper that is citing those papers.

Tang et al [4] and Zhang et al [43] used self-citations to investigate whether the citing and cited publications belonged to the same author. Their work is similar to ours in a sense that they use references (self-citations). They consider two papers authored by the same person if one of them cites the other. We, on the other hand, compare all references

of the two publications. So our work is totally different from their work. Aljaber et al [116] exploit reference markers contexts to estimate similarity between the two publications. Their approach scans the whole script to find reference markers contexts. These contexts are then compared to estimate the similarity. Their work is different from ours that they compare reference markers contexts while we compare the ref-titles and ref-coauthors of all references. The reference markers contexts may or may not represent the cited work properly as every writer describes the cited work in his/her own style and according to the flow and the need of the paper. Two reference marker contexts of the same work by two different authors may have totally different wordings. We, in Shoaib et al [23], investigate the importance of the titles and co-authors of references. This work reveals that the references can be used as sources of information for academic publication data. To the best of our knowledge this is the first work to use references for estimating publications similarity.

2.4. Summary of AND Works

We summarize the characteristics found in AND methods in table 4. Columns two to five show similarity functions, evidences (attributes), clustering/classification techniques and datasets exploited by respective works. Column six speaks about the performance or evaluation measurements. The last column describes sub problems resolved by the respective works. Here, in column six, we restrict AND problems only in two broader categories, i.e., polysemy and synonymy.

Table 4: Summary of the AND approaches

Ref#	Similarity function	Evidence	Clustering/ classification technique	Collections	Evaluation metric	Sub problem
[29]	Larned through Naïve Bayes and SVMs	Citation attributes	SVM and naïve Bayes classifiers	DBLP and Web	Accuracy	Both
[9]	Cosine	Citation attributes	Spectral clustering	DBLP and Web	Accuracy	Both
[49]	Euclidean, distance	EM Citation attributes	Hierarchical naïve Bayes with	DBLP and Web	Accuracy	Both
[15]	LDA-ER	Author names	LDA with Gibbs sampling	CiteSeer and arXiv	F1	Both

[71]	Learned using LASVM	First page of the articles	DBSCAN	CiteSeer	Pairwise F1	Both
[72]	Quasi-clique	Citation/Movie attributes		ACM, BioMed and IMDb	Ranked recall and precision	Syno-nymy
[51]	Levenshtein and Euclidean distance	Citation attributes and latent topics (LDA/PLSA)	Agglomerative	CiteSeer and Web	Pairwise and cluster F1	Both
[52]	Common neighbours, Jaccard, Adamic/Adar and Higher-order neighbourhoods	Author name	Agglomerative	CiteSeer, arXiv and BioBase	F1	Both
[54]	Probabilistic metric	Citation attributes, email, address, keywords and research field	Agglomerative	ISI-Thomson		Both
[31]	Error-drive and hank-based learning	All of each collection	Agglomerative	DBDL and Rexa	Pairwise F1, MUC and B-Cubed	Both
[55]	Learned using SVM	Citation attributes, topics and Web pages	Partitioning	DBLP	Accuracy, Precision and Recall	Both
[14]	Learned using C4.5/SVMs and edit distance	Citation attributes,	Spectral and Agglomerative clustering	DBLP	Pairwise F1	Both
[17]	Learned using Random Forest Classifier	MEDLINE metadata		MEDLINE	Accuracy	Both
[56]	Heuristic	Author names and Web pages	Agglomerative	Korean citations	F1 and under/ over-clustering error	Poly-semy
[57]	Heuristic	Citation attributes	Agglomerative	DBLP	Pairwise and cluster F1and K	Both
[21]	Learned a probabilistic metric	MEDLINE metadata	Agglomerative	MEDLINE	Recall	Both
[22]	Cosine and learned from examples	Citation attributes	Associative classifier	DBLP and BDBComp	Pairwise F1 and K	Both
[65]	Cosine, Semantic relation through	Citation attributes, affiliation,	graphs, longest cycle detection	DBPL	Precision, recall, f-	Both

	social networks	email, topics	algorithm		measure	
[19]	graph-based	Author names	Affinity Propagation	DBLP and MEDLINE	Pairwise F1	Poly-semy
[4]	Cosine, contrastive divergence algorithm, graph, Gibbs sampling,	Publication attributes, relational attributes	HMRFs, graph partitioning method,	ArnetMiner	Pairwise precision, recall and f-measure	Both
[118]	Fragmented comparison and Cosine	Citation attributes	Agglomerative	DBLP and BDBComp	Pairwise F1	Both
[119]	Learned using maximum entropy or logistic regression	Citation attributes and Web pages	Partitioning	DBLP, Penn and Rexa	Accuracy and Pairwise F1	Both
[120]	Social network measures	Citation attributes		DBLP, Cora and BDBComp	F1	Both
[121]	Association rules	Citation attributes	Associative classifier	DBLP and BDBComp	F1	Both
[48]	Cosine, Content and relational based similarity	citation attributes	Hidden Markov Random Fields	ArnetMiner	Pairwise F1	Poly-semy
[18]	Cosine	Citation attributes, abstract, author email & affiliation, topics		DBLP	Precision, recal F-measure	Poly-semy
[87]	Heuristics, weights of common features	Citation attributes	Heuristic based deterministic clustering	PubMed	Accuracy	Both

In this chapter we have described the related research work in detail by categorizing previous works in different categories. The specific research work related to each problem is provided in respective chapters (from chapter 3 to 5).

2.5. Problem Definition

In above discussion we have described AND approaches, similarity measures and sources of information. Here we describe the problem we have addressed in subsequent chapters.

Suppose we have q number of authors that share an ambiguous author name A. We can represent this as $A=\{a_1, a_2, \ldots, a_q\}$, where q is the number of individual authors

belonging to *A*. Again, suppose that we have a set of citations $C = \{c_1, c_2, \ldots, c_z\}$, where z is the number of citations, all sharing author name *A*.

Given a set of citations $C = \{c_1, c_2, \ldots, c_z\}$ *sharing same ambiguous author name A, group C into q disjoint clusters* $K = \{k_1, k_2, \ldots k_q\}$ $(1 \leq q \leq z)$ *such that citations within each cluster* k_i *belong to the same author* $a_i \in A$*, and no citation* c_i *is member of any two clusters, i.e.,* $k_i \cap k_j = \phi$.

In bibliographic databases the number of actual authors sharing an ambiguous name *A*, or the number of clusters (*q*) is unknown [35]. This makes author name disambiguation more problematic [35].

2.6. Chapter Summary

In this chapter we have summarized AND research works by categorizing them in different categories, i.e., machine learning, non machine learning, Statistical relational learning, graph-based and ontology-based. We have also compared and contrasted our work with the works those are similar to ours'. In Table 1 we have provided summary of different approaches. This table mentions the clustering/classification algorithms, similarity functions, information sources (attributes), datasets and sub problems resolved. We have also reviewed the similarity measures used in AND process along with the information sources. We have also provided a table. At the end we formulize the AND problem and provided the problem definition in section 2.5.

Chapter 3. Improving Similarity Measures for Publications

Chapter 3

Improving Similarity Measures for Publications

In this chapter we propose different similarity measures. Each similarity measure is useful for different types or nature of data. Our main focus in this chapter is to devise such measures that can depict our assumptions and help improve the disambiguation process.

3.1. Introduction

In many real life text mining applications such as clustering documents and author name disambiguation (AND) similar documents are grouped together by estimating similarity among them in pair wise fashion. In literature many similarity measures such as Jaccard coefficient, cosine and Dice coefficient are proposed for the comparison of two publications. Out of similarity measures discussed in literature cosine similarity is the most popular metric for measuring document similarity [107] [100]. In document clustering works like [104] [108] and in author name disambiguation works like [29] [22] exploit cosine measure. In this chapter we propose different similarity measures for different types of attributes. In following subsection we also introduce few popular similarity measures used in AND process and document clustering.

3.1.1. Existing Similarity Measures

Similarity functions such as cosine, Dice, Jaccard base on VSM [37]. In VSM documents are represented as vectors of documents, and term weights are calculated by TFIDF [75]. A vector similarity function is used to compute the similarity between vectors. Now we briefly describe the similarity functions that we have used as baselines to compare our proposed similarity measure.

3.1.1.1. Cosine Measure

Cosine similarity is the most popular measure [104] for estimating document similarity based on VSM. The similarity between two documents d_i and d_j can be defined as the normalized inner product of the two corresponding vectors $\mathbf{d_i}$ and $\mathbf{d_j}$[25] and is given in equation 2

$$Sim\left(d_i,\ d_j\right) = \frac{\mathbf{d_i}.\mathbf{d_j}}{|\mathbf{d_i}|\mathrm{X}|\mathbf{d_j}|} = \frac{\sum_{t_x \epsilon u}(w_{t_x}(d_i)\ \mathrm{x}\ w_{t_x}(d_j))}{\sqrt{\sum_{t_x \epsilon d_i} w_{t_x}^2(d_i)\ \mathrm{x} \sum_{t_x \epsilon d_j} w_{t_x}^2(d_j)}} \quad (2)$$

Where $u = (d_i \cap d_j)$ i.e., common terms of documents d_i and d_j; $w_{t_x}(d_i)$ and $w_{t_x}(d_j)$ are the weights of term t_x in documents d_i *and* d_j respectively.

3.1.1.2. Dice Measure

Dice similarity measure can be defined as[26]:

$$Sim\left(d_i,\ d_j\right) = \frac{2\ \mathrm{x} \sum_{t_x \epsilon u}(w_{t_x}(d_i)\ \mathrm{x}\ w_{t_x}(d_j))}{\sum_{t_x \epsilon d_i} w_{t_x}^2(d_i) + \sum_{t_x \epsilon d_j} w_{t_x}^2(d_j)} \quad (3)$$

All symbols mean the same as they are in cosine measure.

3.1.1.3. Jaccard Measure

The Jaccard similarity measure can be defined as follows:

$$Sim\left(d_i,\ d_j\right) = \frac{\sum_{t_x \epsilon u}(w_{t_x}(d_i)\ \mathrm{x}\ w_{t_x}(d_j))}{\sum_{t_x \epsilon d_i} w_{t_x}^2(d_i) + \sum_{t_x \epsilon d_j} w_{t_x}^2(d_j) - \sum_{t_x \epsilon u}(w_{t_x}(d_i)\ \mathrm{x}\ w_{t_x}(d_j))} \quad (4)$$

All symbols mean the same as they are in cosine measure.

3.1.1.4. Information Theoretic

Aslam and Frost [122] develop an information-theoretic measure for pair-wise document similarity and is given as follows:

$$Sim\left(d_i,\ d_j\right) = \frac{2\ \mathrm{x} \sum_{t_x \epsilon u} \min\{O_{t_x}(d_i),\ O_{t_x}(d_j)\} \log\ \pi(t_x)}{\sum_{t_x} O_{t_x}(d_i) \log\ \pi(t_x) + \sum_{t_x} O_{t_x}(d_j) \log\ \pi(t_x)} \quad (5)$$

[25] Bold face letters represent vector form of a document.

[26] There exist different formats of Dice and Jaccard measures. We took the definitions of these measures from the work of Wan [103].

$O_{t_x}(d_i)$ is the fractional occurrence of term t_x in document d_i ; and probability $\pi(t_x)$ is the fraction of the corpus (collection) documents containing term t_x.

3.1.2. Terminology, Assumptions and Definitions

In this section we explain the terminology and assumptions used in this chapter. Table 5 lists few names to be referred from here onward for explanation of name similarity.

Table 5: Names and notations used for explanation

Names	Notations	Names	Notations
Muhammad Shoaib	$ň_1$	Shoaib Muhammad	$ň_4$
M. Shoaib	$ň_2$	M. Shoaib	$ň_5$
M. Shoaib kamboh	$ň_3$	M. Shoaib kamboh	$ň_6$
M. Safdar Kamboh	$ň_7$		

3.1.2.1. Types of Tokens in Names

We estimate name similarity by matching name tokens. We divide name tokens into five categories.

Full match token: if a token *t'* from name $ň_i$ is not abbreviated and matches exactly to any of the tokens from name $ň_j$. For example, in table 5, token t'_1 of name $ň_1$ fully matches to the token t'_2 of name $ň_4$.

Abbreviation match token: if a token *t'* of name $ň_i$ is abbreviated and matches exactly to any of the tokens of name $ň_j$. For example, in table 5, token t'_1 ("M.") of name $ň_2$ exactly matches to token t'_1 of name $ň_5$.

Abbr-initial match token: if an abbreviated token *t'* of name $ň_i$ matches to the initial letter of any of the tokens of name $ň_j$ ignoring dot (.) of abbreviated token. For example, token t'_1 of name $ň_2$ ("M.") matches initial letter of t'_1 of $ň_1$ ("Muhammad").

Missing token: a token *t'* of name $ň_i$ may not have any matching token in name $ň_j$. It is possible if and only if $ň_i$ and $ň_j$ have different number of tokens. Consider names $ň_2$ and $ň_6$ in table 5. Names $ň_2$ and $ň_6$ have two matching tokens, but $ň_2$ does not have any token to be matched with a third token (Kamboh) of $ň_6$. This is the case of missing token.

Conflicting token: if two names $ň_i$ and $ň_j$ have a non matching token other than missing token. For example, in table 5, token t'_2 in mane $ň_3$ does not match to any of the tokens in

name $\check{n}_7$. Similarity token t'_2 of mane $\check{n}_2$ does not match to any of the tokens in name $\check{n}_7$. Missing and conflicting tokens are different from each other and they should be treated differently. Missing tokens case occurs only when number of tokens in two names is unequal whereas conflicting tokens case is irrespective of this condition.

3.1.2.2. Assumptions and Definitions

Assumption I

The probability that two names ($\check{n}_i$ and $\check{n}_j$) sharing full matching tokens belong to the same person is higher than that of those sharing abbreviated tokens or abbr-initial tokens. Similarly, the probability that two names sharing abbreviated matching tokens belong to the same person is higher than that of those sharing abbr-initial tokens.

To provide the proof of this assumption, we consider Table 27 in Appendix B. There are 22 unique authors sharing ambiguous name "R. Silva". The author of publication number 26 (R Da Silva) has one abbreviated token, i.e., "r". It may stand for any name token starting from "r". In Table 27 there are two other records (9 and 25) whose author name matches (abbr-initial match) to that of 26's. According to these data, the probability that two citations having such co-author names and chosen at random belong to the same person is 0.33. On the other hand, the co-author of publication number 25 (Roberto Da Silva) has no abbreviated token. Although this name does not have any exact match yet it has one abbr-initial match (the last record). Interestingly these both records belongs to the same author (Author_ID = 23). If we choose these names and estimate the probability whether they belong to the same author, the answer is yes with high probability (in this case 1). This discussion proves assumption I.

Assumption II

The probability that two names $\check{n}_i$ *and* $\check{n}_j$ *having more number of matching tokens belong to the same person is higher than that of those having less number of matching tokens provided they don't have conflicting tokens.*

To provide the proof of this assumption, we consider Table 27 in Appendix B. There are 22 unique authors sharing ambiguous name "R. Silva". There are six such names that consist of 5 tokens. Out of these names, "Ricardo M D A Silva" occurs three times, and

all its occurrences belong to the same author. Any two occurrences of "Ricardo M D A Silva" chosen at random will always refer to the same author. In other words, the probability that any two occurrences of this name chosen at random will belong to the same author is 1.

Again consider Table 27, there are five occurrences of two names "Ricardo M D A Silva" and "Ricardo M A Silva" belonging to three different authors. These five occurrences of the two names share at least four tokens. So any two occurrences of these names chosen at random may or may not refer to the same author. In other words, the probability that any two occurrences of such names chosen at random will belong to the same person is 4/10 = 0.4.

The above discussion provides the proof of assumption II.

Definition 1

Two co-authors ca_i and ca_j *from two different citations* c_i *and* c_j *are considered common if they have name similarity > threshold.*

Assumption III

The probability that two citations c_i*, and* c_j *having more number of common co-authors are from the same person is higher than that of those having less number of common co-authors irrespective of the proportion of common co-authors.*

To provide the proof of this assumption, we consider Table 28 in appendix B. Three pairs of publications ((4, 5), (10, 16) and (15, 16))[27] have at least one common co-author. Out of these, the publication pair (10, 16) is not from the same author. So, according to these data, the probability that two citations having at least one common coauthor are written by the same author is 0.66. On the other hand we could not find even a single pair of citations having at least two common co-authors authored by two different persons having the same name. So, the probability that two citations having at least two common co-authors are authored by the same person is necessarily greater than the previous case. Similarly, the probability that two citations having at least three or more common co-

[27] These numbers are the publication number in Table 28

authors are authored by the same person is necessarily greater than the previous case. This discussion proves that our assumption is based on the real data.

According to assumption III, *co-authors similarity value of two citations c_i, and c_j having higher number of common co-authors should be higher than that of those having less number of common co-authors irrespective of the proportion of common co-authors.*

Definition 2

Two names $\check{n}_i$ and $\check{n}_j$ are identical if and only if they have equal number of tokens and each token in $\check{n}_i$ exactly matches to one of the tokens in $\check{n}_j$ provided $\check{n}_i$ and $\check{n}_j$ don't have any abbreviated token.

According to this definition, "Muhammad Shoaib" is identical to "Muhammad Shoaib" and "Shoaib Muhammad" but "M. Shoaib" is not identical to "M. Shoaib" or "Shoaib M." because M. in $\check{n}_1$ and $\check{n}_2$ may stand for different names.

Definition 3

Two names $\check{n}_i$ and $\check{n}_j$ are compatible if and only if they are neither identical nor have any conflicting token and have name similarity greater than threshold. Compatible names may have missing tokens, but don't have any conflicting token.

According to this definition "Muhammad Shoaib" may be compatible to "M. Shoaib", "Shoaib M.", "Muhammad S.", "M. Shoaib Kamboh", "M. S. Kamboh", etc., but "M. Shoaib" is not compatible to "Muhammad Shahid" or "M. Shahid". Out of these names which are compatible and which are not? It depends upon the threshold user defined value.

Definition 4

Two names $\check{n}_i$ and $\check{n}_j$ are common if and only if they are either identical or compatible.

3.2. Problems Definitions

In this chapter, we focus multiple problems, and here we describe them one by one.

Problem I

The problem is defined in following lines:

Given two names $\check{n}_i$ and $\check{n}_j$ devise a similarity measure that can assign different weights to different types of tokens, and assign higher similarity value for two names $\check{n}_i$ and $\check{n}_j$ if they share more number of tokens than those which share less number of tokens.

Problem II

Given a citation pair $(c_j, c_j) \in C$, devise a similarity measure that can assign more similarity value if (c_j, c_j) pair shares more number of common co-authors as compared to the pair (c_j, c_j) sharing less number of common co-authors

Problem III

Jaccard coefficient may output similarity greater than 1 for two titles T_1 and T_2 if they share a common word and one of them has its frequency greater than 1. For example, consider two synthetic titles "Mr. Books and Books: Books" as T_1 and "Mr. Books Property" as T_2. Jaccard coefficient outputs title similarity for T_1 and T_2 greater than 1 (4/3.5= 1.14)[28]. In a small document, though the chances that a word is repeated are very low yet not zero. Similarity values generated by cosine are leaner but not proportional to the number of common words provided that the total number of words in both titles remain same.

Given two short textual documents, devise a similarity measure that can produce similarity values proportional to the common and total number of words and restrict the similarity output between 0 and 1.

Problem IV

In two ref-titles, a word may repeat itself many times in the first one and may appear only once in second one. Our proposed equation 12 and cosine are not the best solution for such type of documents. They do not assign additional weight to a common word if its frequency is different in both ref-titles. Further, cosine assigns more weight to less frequent non common words. Equation 12 assigns proper weight to non common words, but it doesn't care for frequency of non common words. Cosine may produce inverse

[28] We consider each repeating word of T_1 or T_2 as common with T_2 or T_1 if it exits in T_2 or T_1. For example the term "Mr." occurs only once in both titles, and the term "Books" is repeated three times in T_1 and only once in T_2. We have 1+3 = 4 common terms and 7 total terms. In above example "and" is not counted and it is considered as stop word.

trend of similarity values for documents (ref-titles) if we go on changing common words frequencies in one document without changing the number and frequencies of non common words.

Given two long textual segments (say, ref-titles) that may repeat a term t an arbitrary number of times devise a measure that can output similarity value very close to absolute value and can streamline the similarity value giving appropriate weights to non-repeating (frequency =1), repeating (frequency >1), common and non common words.

In problem III and IV, we are concerned to estimate syntactic similarity only, and not the semantic similarity. Syntactic similarity methods (VSM based Cosine, Jaccard) are those which compare two documents blindly and are unaware of the context and semantics of the word used. On the other hand, semantic similarity approaches such as topic modeling methods [102] [44] or ontological [97] [123] or WordNet based techniques [113] are aware of the meanings and context of the word used.

3.3. Proposed Similarity Measures

In this section we explain our proposed similarity measures.

3.3.1. Name Similarity

Name similarity between the two names is useful in blocking step[29] as well as in estimating co-authors similarity. Suppose we have two sets of names N1 and N2 such that:

N1 is the number of co-author names in one of the citations of set *C* and

N2 is the number of co-author names in another citation of set *C.*

Where $C = \{c_1, c_2, \ldots, c_z\}$.

To estimate similarity between two names $\check{n}_i$ and $\check{n}_j$ the similarity measure based on Jaccard formula is given in equation 6.

[29] In this step, in AND process, citations of compatible names are grouped together to avoid unnecessary comparisons between citations of non compatible names. These groups are called ambiguous groups.

$$Sim_{nam}\,(\check{n}_i, \check{n}_j) = \frac{(e * \alpha + b * \beta + q * \gamma)}{(\check{z} * 0.5 + h * 100)} \quad (6)$$

Where $(\check{n}_i, \check{n}_j) \epsilon (N1, N2)$; α, β and γ are constant weights of tokens of type e (exact matching), type b (abbreviation matching) and type q (abbr-initial matching) respectively; e , b and q represent number of exact matching tokens, abbreviation matching and abbr-initial matching tokens in $(\check{n}_i, \check{n}_j)$ pair respectively; and h and ž are the number of conflicting and total number of tokens in both names $\check{n}_i$ and $\check{n}_j$. In above equation $h * 100$ factor decreases the similarity value of two different names (having conflicting tokens) near to 0.

Why do we assign different weights to different types of tokens? Consider name similarities in table 6 estimated through equation 6 with homogenous weights (i.e., 1), and variant weights (1, 0.95, 0.90 for e, b and q respectively). Homogenous weighting scheme estimates same similarity value (i.e., 1) for all pairs of names in table 6. Is it rational to say Sim(Ali Daud, Ali Daud) = Sim(A. Daud, A. Daud) = Sim(A. Daud, Ali Daud)? The probability of two names in record 1 (of table 6) being the same person is greater than that of 2's[30]; and record 2's probability is greater than that of 3's. So Sim(Ali Daud, Ali Daud) > Sim(A. Daud, A. Daud), and Sim(A. Daud, A. Daud) > Sim(A. Daud, Ali Daud). To depict our assumption I we employ variant weighting scheme for different types of tokens. It estimates a higher similarity value for two names of record 1 than that of those in record 2 and 3 (table 6, column 5). Same is true for records 2 and 3.

Table 6: Name similarities estimated through equation **6**

Sr#	Name 1 ($\check{n}_i$)	Name 2 ($\check{n}_j$)	Sim($\check{n}_i$, $\check{n}_j$) with same weights	Sim($\check{n}_i$, $\check{n}_j$) with variant weights
1	Ali Daud	Ali Daud	1	1
2	A. Daud	A. Daud	1	0.975
3	A. Daud	Ali Daud	1	0.95

Equation 6 helps us to assign variant similarity weights to different types of tokens. It holds true for our assumption I but fails to hold true for our assumption II. To depict our assumption II we multiply equation 6 with log factor and get equation 7.

[30] There may be multiple name tokens starting from A.

$$Sim_{nam}\,(\check{n}_i, \check{n}_j) = \frac{(e\ *\ \alpha\ +\ b\ *\ \beta\ +q\ *\ \gamma)}{(\check{z}\ *\ 0.5\ +\ h\ *100)} * log(\check{z} + 2) \quad (7)$$

Here log means log_{10}; $\check{z} \in \{2, 3,\ldots,8\}$ is the total number of tokens in both names. All other symbols are same as they are in equation 6. The term $\check{z} + 2$ should not exceed 10 otherwise this factor may cause Sim ($\check{n}_i$, $\check{n}_j$) > 1. It is assumed, on the base of our observations, that maximum number of tokens in a name are 4 hence $\check{z}$ will not exceed 8 producing $\check{z}$ + 2 =10. The constant 2 in log factor is not compulsory. It provides relatively more proportional logarithmic weight to names having less number of tokens. It is further assumed that name similarity of two names $(\check{n}_i, \check{n}_j)$ having four exact matching tokens should get similarity value 1. This assumption is based on our notion---it is very hard to find a case that two co-authors of two different citations having a common ambiguous author name and sharing 4 exact matching tokens (identical names) represent two different persons.

3.3.2. Co-authors Similarity

To estimate co-authors similarity we may use Jaccard like formula given in equation 8 along with equation 7.

$$\mathrm{Sim}_{\mathrm{CA}}(c_i,\ c_j\ ) = \frac{\Sigma\left(Sim_{nam}(\check{n}_x\ ,\ \check{n}_y)\right)}{\eta * 0.5} \quad (8)$$

Where $(\check{n}_x,\ \check{n}_y) \in (\check{n}_i, \check{n}_j)$ such that $Sim_{nam}(\check{n}_i\ ,\ \check{n}_j) > thr$; $(\check{n}_i, \check{n}_j)$ are same as in equation 2; and η is the total number of names in (c_i, c_j) pair.

On the other hand, Ferreira et al [22] and Cota et al [25] standardize co-author names using only the initial letter of the first name along with the full last name. Standardizing this way may convert two different names to the same name. To avoid such scenarios we estimate co-authors attribute similarity in a different style.

The co-authors similarity of two citations estimated through equation 8 is proportional to the sum of similarity values of common co-authors and the total number of co-authors in c_i and c_j. Output of equation 8 may be reverse to that of our assumption III. To depict our assumption III in co-authors similarity we introduce equation 9.

$$\mathrm{Sim}_{\mathrm{CA}}(c_i, c_j) = log\left(\sum\left(Sim_{nam}(\check{n}_x\ ,\ \check{n}_y)\right) + 1\right) - \frac{\log(\eta' + 1)}{\log(\eta + 1)} * \frac{1}{10 + \eta + 1} \qquad (9)$$

Where $(\check{n}_x, \check{n}_y)$ are same as in equation 8; η' is the number of non common co-authors in (c_i, c_j) pair; η is the total number of names in (c_i, c_j) pair; in the numerator of second term "1" is added to η' to avoid log(0) case. The term "co-authors" is used to represent a single attribute of citations. We use ending "s" of "co-authors" to represent that there are multiple co-authors in one citation.

In equation 9, $log\left(\sum\left(Sim_{nam}(\check{n}_x\ ,\ \check{n}_y)\right) + 1\right)$ causes logarithmic increment to $\mathrm{Sim}_{\mathrm{CA}}(c_i, c_j)$ w. r. t. increasing number of common co-authors; the term $\frac{\log(\eta\prime+1)}{\log(\eta+1)}$ is dissimilarity factor; and the term $\frac{1}{10+\eta+1}$ gives lesser weight to dissimilarity factor as the number of common co-authors increase and vice versa. This equation depicts our assumption III for all observed or expected values of η and η' in AND scenario. Theoretically, this equation may violate assumption III (slightly) only if η-$\eta' > 7$ and η' approaches to a fairly large number. In real life citation datasets it may never violate. Note: equation 9 may output co-authors similarity > 1 if number of exact matching co-authors is beyond 9. It is expected (on the base of our observations) not to happen in real life citations. If this happens, in any scenario, we may consider it "1". Equation 9 outputs a value less than 0 (minimum -0.1) if there is no common co-author in two citations. This similarity value is considered as 0. Thus the value of co-authors similarity remains within 0 to 1 for all expected values of η and η' in a citation dataset. Results show that $\mathrm{Sim}_{\mathrm{CA}}(c_i, c_j)$ estimated through equation 9 depicts assumption III. In real citations, $\mathrm{Sim}_{\mathrm{CA}}(c_i, c_j)$ for any arbitrary number of $\eta - \eta'$ is always greater than that of $\eta - \eta' - 1)$'s.

3.3.3. Short Segment Similarity

We may use Jaccard coefficient for title and venue similarity. According to Jaccard coefficient title and venue similarities can be estimated through equation 10 and 11 respectively:

$$Sim_{title}(c_i, c_j) = \frac{(|t_{c_i}| \cap |t_{c_j}|)}{0.5 * (|t_{c_i}| + |t_{c_j}|)} \quad (10)$$

$$Sim_{ven}(c_i, c_j) = \frac{(|t_{c_i}| \cap |t_{c_j}|)}{0.5 * (|t_{c_i}| + |t_{c_j}|)} \quad (11)$$

Where t_{c_i} and t_{c_j}, are terms (words) in titles or venues of citations c_i and c_j respectively.

Jaccard coefficient (equation 10) may output similarity greater than 1 for two titles T_1 and T_2 if they share a common word and one of them has its frequency greater than 1. For example, consider two synthetic titles "Mr. Books and Books: Books" as T_1 and "Mr. Books Property" as T_2. Jaccard coefficient outputs title similarity for T_1 and T_2 greater than 1 (4/3.5= 1.14)[31]. In a small document, though the chances that a word is repeated are very low yet not zero.

Similarity values generated by cosine based on VSM are leaner but not proportional to the number of common words provided that the total number of words in both titles remain same.

We devise a similarity measure that can compare two small text documents (titles, venues) by matching key words, and produce similarity values proportional to the common and total number of words, and restrict the similarity value between 0 and 1. For this purpose we modify equation 10 as equation 12.

$$Sim_{title}(c_i, c_j) = \frac{(|t_{c_i}| \cap |t_{c_j}|)}{(|t_{c_i}| \cap |t_{c_j}|) + ((|t_{c_i}| \cup |t_{c_j}|) - (|t_{c_i}| \cap |t_{c_j}|)) * 0.5} \quad (12)$$

Where t_{c_i} and t_{c_j} are same as defined in equation 10. Equation 12 outputs title similarity for T_1 and T_2 in the example described above as 8/9 =0.89. We carry on our discussion w. r. t. title attribute only because the estimation of venue attribute is exactly same as that of title's attribute. We name equation 12 as a conditional absolute measure (CAM) because it outputs the absolute similarity value between two documents (titles, venues) provided each word of both documents occurs only once.

[31] We consider each repeating word of T_1 or T_2 as common with T_2 or T_1 if it exits in T_2 or T_1. For example the term "Mr." occurs only once in both titles, and the term "Books" is repeated three times in T_1 and only once in T_2. We have 1+3 = 4 common terms and 7 total terms. In above example "and" is not counted and it is considered as stop word.

CAM has two advantages over VSM based cosine. Its time complexity is low and its output is proportional to the percentage of common data between two citations. CAM has one weakness, i.e., it does not make justice for common words, if their term frequencies are not same in documents to be compared. We ignore this weak point, assuming this situation may not happen in titles and venues of citations. We favor using CAM for titles and venues attributes in place of Jaccard or cosine.

3.3.4. Long Segments Similarity

We, in reference [23], propose employing titles of references of publications (ref-titles) as an attribute to improve the similarity between publications. We combine all titles of references of a publication into one title and name it as ref-titles. If we have *r* references of a publication *p* then there are *r* titles as each reference has exactly one title[32]. Aggregating *r* titles into one title gives us one *ref-titles*. The term "Ref-titles" is used to represent a single attribute. We use ending "s" of "ref-titles" to represent that there are multiple (*r)* ref-titles in one publication. Ref-titles have many repeating terms. Existing similarity measures (especially cosine) do not assign appropriate weights to non-repeating, repeating, common and non common words. In equation 14, we introduce a similarity measure to streamline ref-titles similarity giving appropriate weights to non-repeating, repeating, common and non common words. From here onwards we call this similarity measure as SDK index[33] (Shoaib et al [106]). We derive SDK index from CAM defined in equation 12, and Shoaib index [105] given in equation 13.

$$\mathrm{Sim}_{rt}(p_i, p_j) = \frac{\sum_{t_x \epsilon u}\left(\frac{1}{1+\log\left(\max\left(f_{t_x}(p_i,p_j)\right)/\min\left(f_{t_x}(p_i,p_j)\right)\right)}\right)}{u + 0.5 * u' + \left(\sum_{t_y \epsilon u'} log\left(f_{t_y}(p_i,p_j)\right)\right)} \quad (13)$$

Where rt represents ref-titles attribute; $u = \{p_i \cap p_j)\}$ and $u' = \{p_i \cup p_j) - (p_i \cap p_j)$; $maxf_{t_x}(p_i, p_j)$ is the maximum frequency of term t_x in publication p_i or p_j ; $minf_{t_x}(p_i, p_j)$ is the minimum frequency of term t_x in publication p_i or p_j ; and $f_{t_y}(p_i, p_j)$ is the frequency of term t_y in p_i or p_j.

[32] We ignore the references that don't have titles. E.g., a web page URL may not have a title at all.
[33] Shoaib, Daud and Khiyal, the last name initials of the authors who proposed SDK index

The numerator in equation 13 has been just like CAM with the only difference by introducing term frequency. In denominator u' is multiplied by 0.5 because in the numerator we count two occurrences of a common term (one occurrence in each publication) as one term. The last term in denominator is used to logarithmically decrease the similarity output against frequency of non-common terms. Shoaib index provides proportional weights to a number of common and non common words, assigning logarithmic weights to their frequencies. The reason to provide logarithmic weight is that if a word is repeated ten times in a document (ref-titles) it is appropriate to say that it is ten times more important (similar or dissimilar), and it is also unfair to ignore the frequency at all.

Shoaib index is relatively away from absolute measure for higher difference in term frequencies. In other words its output is not close enough to absolute value when ratio between the frequencies of terms in two publications p_i and p_j goes far beyond 1. To be (compared) as close to absolute value as possible, we add the sum of logarithmic squares of differences in the frequencies of common words to the denominator of Shoaib index and get SDK index (equation 14).

$$\mathrm{Sim}_{rt}(p_i,\ p_j) = \frac{\sum_{t_x \epsilon u}\left(\frac{1}{1+\log\left(\max\left(f_{t_x}(p_i,p_j)\right)/\min\left(f_{t_x}(p_i,p_j)\right)\right)}\right)}{u+0.5*u'+\sum_{t_x \epsilon u}(log\ (maxf_{t_x}(p_i,p_j)-\ minf_{t_x}(p_i,p_j)))^2+(\sum_{t_y \epsilon u'} log\ (f_{t_y}(p_i,p_j))} \quad (14)$$

SDK index provides proportional weights closer to the absolute value than other measures. SDK index needs not any information about collection of documents. In other words, it is independent of the number of documents in the collection. It needs just information of the two documents to be compared. SDK index is basically designed for textual documents. It can also be applied for co-authors and ref-coauthors along with equation 7 and 9.

3.4. Results and Discussion

Here we consider synthetic examples and data to prove that our proposed measures are closer to the assumptions defined in this chapter. We have shown different trends of similarity outputs by varying inputs in a sequential style. To have such analysis of

original data is very difficult (perhaps impossible). The impact of these similarity measures in AND is shown in chapter 5.

3.4.1. Name Similarity

Consider the examples in table 7. It is more probable that two names of record 3 belong to the same person than those of record 2's and 1's. While estimating name similarities we assign weights 1.0, 0.95 and 0.9 to exact, abbreviation and abre-initial matching tokens respectively. Jaccard based coefficient or equation 6 estimates name similarity in reverse order to that of our assumption II (table 7, column 4). Table 7 shows name similarities estimated through equation 7 with log factor. It is clear that equation 7 assigns more similarity value to names having more number of matching tokens than those having less number of matching tokens. For example, names in record 3 are more similar than those in record 2. The same is true for records 3 and 1, and records 2 and 1. Equation 7's similarity estimations hold true for our assumption II.

Table 7: Name similarities estimated through equations **6** and **7**

Sr #	**$\check{n}_i$ (Name 1)**	**$\check{n}_j$ (Name 2)**	**Sim($\check{n}_i$, $\check{n}_j$) through Jacc. based Coef. (eq. 6)**	**Sim($\check{n}_i$, $\check{n}_j$) through eq. 7 with log($\check{z}$)**	**Sim($\check{n}_i$, $\check{n}_j$) through eq. 7 with log ($\check{z}+1$)**	**Sim($\check{n}_i$, $\check{n}_j$) through eq. 7 with log($\check{z}+2$)**
1	M. Shoaib	M. Shoaib	0.975	0.587	0.699	0.759
2	M. S. Kamboh	M. S. Kamboh	0.967	0.752	0.817	0.873
3	M. S. H. Khiyal	M. S. H. Khiyal	0.963	0.869	0.918	0.9625

3.4.2. Co-authors Similarity

The co-authors similarity of two citations estimated through equation 8 (Jaccard like coefficient) is proportional to the number of common co-authors to the total number of co-authors in $\underline{c_i}$ and c_j. The output of Jaccard like coefficient may be reverse to that of our assumption III. For explanation, consider synthetic citations in table 8 and their co-authors similarities in table 9. In these citations "M. shoaib" or "Muhammad Shoaib" is the ambiguous author name. We do not include this name in co-authors, while estimating co-authors similarity. Citations c_1 and c_2 have only one common co-author (Ali Daud).

Table 8: Synthetic citations dataset

Cit. #	Co-authors of citations	Titles and venues etc. of citations
c_1	**Muhammad Shoaib,** Ali Daud	"Role of references in documents similarity estimation," *Journal of Information Systems*, 9(3), pp. 222-245, 2011.
c_2	**M. Shoaib**, Ali Daud	"Document similarity estimation through references," *International Journal of Information Engineering*, 20 (4) pp. 111-145, 2010.
c_3	**M. Shoaib**, Ali Daud, Hikmat Khan	"Document clustering though references," In proc. of IEEE conf. on txt mining, pp. 100-110, 2012 .
c_4	**M. Shoaib**, Ali Daud, Hikmat Khan, Malik Sikandar Hayat Khiyal	"Hierarchical clustering for name disambiguation in digital libraries," *Journal of Information Sc. and Digital Library* , 15(2), pp. 200-225, 2010
c_5	**M. Shoaib**, Ali Daud, Hikmat Khan, Malik Sikandar Hayat Khiyal, Ali Ahmad	"Unsupervised hierarchical clustering for name disambiguation in bibliographic databases ," *Journal of Information Science* ,15(2), pp. 200-225, 2011
c_6	**M. Shoaib**, Ali Daud, Hikmat Khan, Malik Sikandar Hayat Khiyal, Aneel Rahim	"Un-supervised hierarchical clustering for name disambiguation in bibliographic databases," *International Journal of Information Science*, 18 (1) pp. 99-125, 2013.
c_7	**M. Shoaib**, Ali Daud, Hikmat Khan, Malik Sikandar Hayat Khiyal, Aneel Rahim, Zeshan Shafi, Imran Razzaq, Adil Badar, Asad Meer, Imran Saeed	"Supervised clustering for name disambiguation" *Science Journal*, 18 (1) pp. 99-125, 2013.

Table 9: Comparison of Jaccard like coefficient and proposed co-authors similarity measure (equation **9**)

Sr#	Citation pairs (c_i, c_j)	$\mathbf{Sim_{CA}}(c_i, c_j)$ **through eq.** 8 **(Jaccard like coef.)**	$\mathbf{Sim_{CA}}(c_i, c_j)$ **through eq.** 7 **and** 9
1	(c_1, c_2)	0.778/2*0.5 = 0.778	log(1.778)-0= 0.25
2	(c_3, c_4)	1.556/5*0.5 = 0.623	log (2.56)-0.024= 0.384
3	(c_5, c_6)	2.556/8*0.5= 0.639	log(3.56)-0.026=0.525
4	(c_6, c_7)	3.334/13*0.5= 0.513	log(4.33)-0.044=0.592

Table 9 (column 3) shows that $\mathrm{Sim_{CA}}(c_6, c_7) < \mathrm{Sim_{CA}}(c_5, c_6) < \mathrm{Sim_{CA}}(c_1, c_2)$. Whereas, according to our assumption III, it is more probable that c_6 and c_7 are from the same "M. Shoaib" as compared to that of c_5 and c_6 or c_3 and c_4 or $c_1$and c_2. Table 9 (column 4) shows that our equation 9 is in accordance with assumption III as it assigns more similarity value ($\mathrm{Sim_{CA}}(c_i, c_j)$ to the citations sharing a number of common co-authors irrespective of their proportion of common co-authors.

3.4.3. Short Segment Similarity

Table 10 shows similarities[34] of the title attribute of citations of table 8 estimated through cosine and equation 12 (CAM).

Table 10: Comparison of cosine and CAM

Sr#	Citations pairs (c_i, c_j)	VSM based Cosine	CAM
1	(c_5, c_6)	1.0	1.0
2	(c_1, c_2)	0.594	4/4.5 = 0.889
3	(c_2, c_3)	0.160	2/3.5 = 0.571
4	(c_3, c_4)	0.041	1/4.5 = 0.222

Table 10 shows that CAM outputs are absolute values. For explanation consider table 11 and figure 3. In table 11, we took two text documents of length 10 words each, and each word's frequency in each document was at the most one, and then we estimated similarities by varying number of common words from 1 to 10.

Table 11: Comparison between cosine and CAM for different percentages of common data

Sr#	% of Common Words	VSM based cosine	Equation 12 (CAM)	Sr#	% of Common Words	VSM based cosine	Equation 12 (CAM)
1	0	0	0	7	60	0.170	0.6
2	10	0.015	0.1	8	70	0.241	0.7
3	20	0.033	0.2	9	80	0.353	0.8
4	30	0.055	0.3	10	90	0.551	0.9
5	40	0.083	0.4	11	100	1.0	1.0
6	50	0.120	0.5				

Table 11 and figure 3 clearly show that CAM is an absolute measure, whereas VSM based cosine is not. VSM based cosine and CAM both assign absolute similarity values (i.e., 1 or 0) when both documents share either 100% or 0 % data. For all other cases cosine does not assign absolute similarity value, whereas CAM assigns absolute similarity values for all cases. CAM has two advantages over cosine. Its time complexity is low and its output is the absolute value as shown in figure 3.

[34] Similarities were estimated after stemming and stop-word removing in both methods

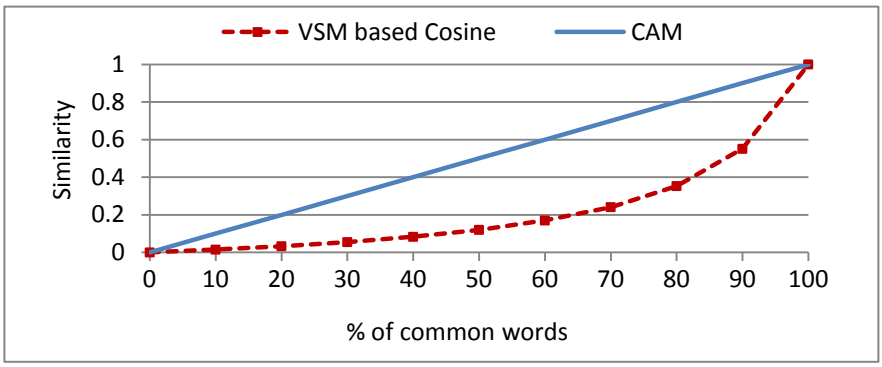

Figure 3: Comparison between cosine and CAM for different % of common data

3.4.4. Long Segment Similarity

Here we consider synthetic examples and data to prove that SDK index is closer to the assumptions than baseline methods. We show different trends of similarity outputs by varying inputs in a sequential style. To have such analysis of real data is very difficult (perhaps impossible). We compare SDK index with Shoaib index, Cos-TFIDF, Cos-TF, Dice coefficient, Jaccard coefficient and information theoretic. For all measures except Cos-TFIDF we weigh the term frequency by log (TF). To avoid the possibility of log (0) case we weigh as log (1+TF). We implemented these measures in MS Excel 2007. We compare and explain the behavior of different similarity functions, especially focusing cosine measure in following four scenarios.

3.4.4.1. Scenario I

The effect of frequency difference of common words when there is no non common word: We take two synthetic documents d_1 and d_2 having a single common word between them and no non common word. We go on increasing the frequency of (common) word in the document d_2 from 1 to 20 remaining the document d_1 unchanged. Figure 4 illustrates trend of different similarity functions for this scenario.

Figure 4 shows that Cos-TFIDF and Cos-TF are not suitable for this scenario because they are not affected by the frequency difference of common words. All other measures show a linear trend as they decrease the similarity values when the frequency ratio of common words in both documents goes beyond 1. The SDK index curve is more affected than other measures for all values of frequency difference of common words. Thus, it can be considered closer to the absolute similarity value.

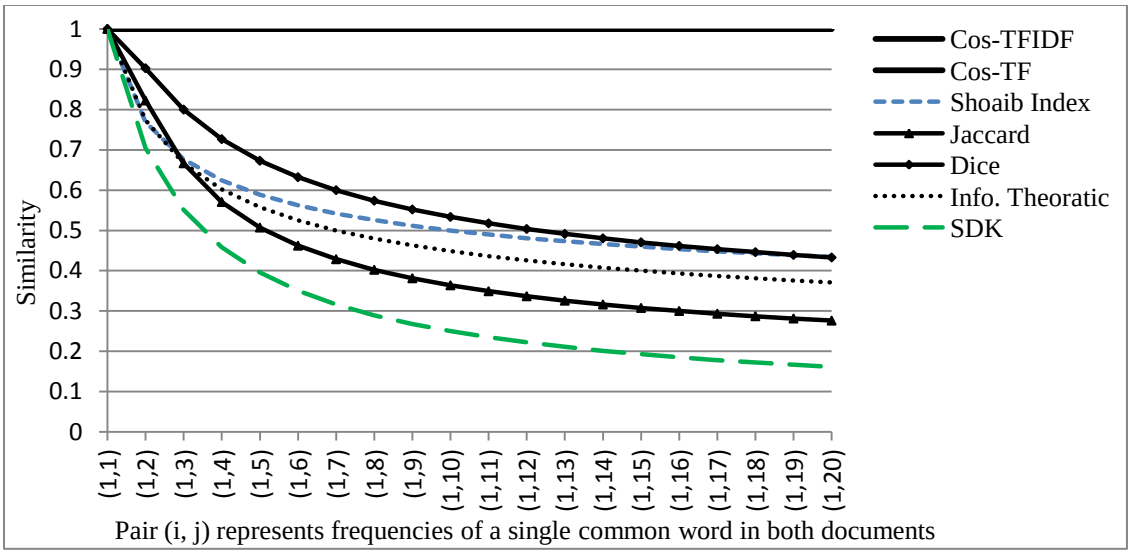

Figure 4: Effect of frequency difference of common words when there is no non common word

3.4.4.2. Scenario II

Effect of number of non common words: We take two documents having two common words and, initially, no non common word. We go on increasing the number of non common words each having frequency 1 in both documents alternatively. Figure 5 demonstrates the effect of a number of non common words.

Figure 5 shows that Cos-TFIDF is much affected for smaller values of the number of non common words and little (negligible small) for such larger values. SDK index, Shoaib index, Dice and information theocratic all are same in this scenario. Cos-TF differs from these measures, but the difference is negligible small[35]. SDK index, Shoaib index, Dice and information theocratic assign proportional weight to common and non common words. It is clear that when the number of non common word changes from 0 to 16 (from 0% to 80%) the similarity value changes with the same ratio (percentage). In other words, they all are absolute measures in this scenario. For example, similarity value is 0.5 when the percentage between common and non common data is 50. Cosine curve does not have this beauty. Jaccard's output is in between SDK index and Cos-TFIDF.

[35] For example, when documents d_1 and d_2 have two common words, and document d_1 has one non common word, this difference is 0.016496581.

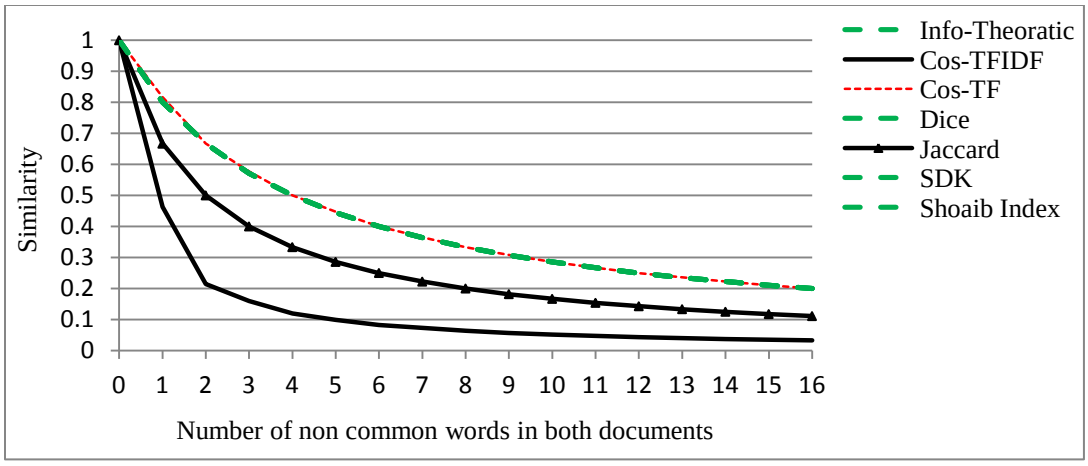

Figure 5: Effect of number of non common words

3.4.4.3. Scenario III

Effect of frequency difference of common words when non common words also exist: We take two documents d_1 and d_2 having ten common and six non common words, initially all words having frequency 1. Each time we increase the frequency of each common word in document d_2 by 1 without changing the document d_1 and non common words. Out of six non common words, three are in the document d_1 and three in d_2. Figure 6 depicts this scenario. This scenario is different from scenario I. In scenario I documents d_1 *and* d_2 have no non common words. In this scenario documents also have non common words.

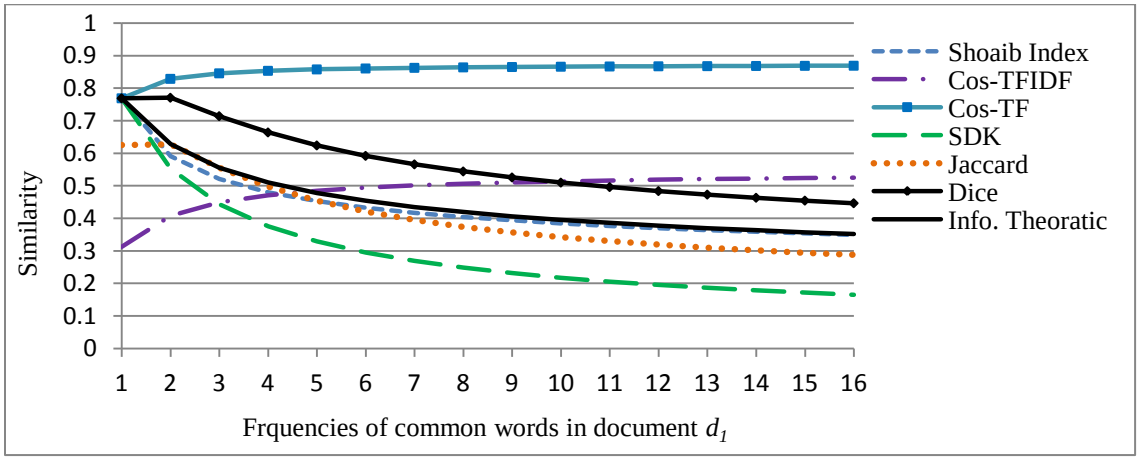

Figure 6: Effect of frequency difference of common words when non common words also exist.

Figure 6 shows that Cos-TFIDF and Cos-TF curves show positive trends where as it should be negative. In the above scenario, for higher proportional frequency difference of common words without any change in non common word similarity values should be

lower. All other measures in figure 6 follow negative trend. Jaccard and Dice coefficients initially (from 1-2 on y-axis) are unchanged, but for all other values they show the same trend as others except cosine. SDK index is the lowest but never reaches to zero. Being the lowest is not guaranteed to be more suitable. Actually SDK index assigns lesser weights to lower values of proportional frequency difference of common words in both documents, and it is also comparatively more fare for such higher values than other measures. We can say that SDK index is nearer to absolute value in this scenario.

3.4.4.4. Scenario IV

Effect of existence of non common words either in both documents or only in one document: this scenario is elaborated in following two cases.

- All non common words are evenly distributed in both documents (e.g., if d_1 and d_2 have 10 non common words: 5 are in document d_1 and five are in d_2. Here we ignore their frequency for simplicity)
- All non common words exist only in the document d_1 or d_2.

Figure 7 depicts this scenario. Figure 7 is drawn considering the data of figure 5. Here we investigate the behavior of different measures. In figure 7, Cos TFIDF and Cos TF illustrate the first case, and Cos TFIDF* and Cos TF* represent second case. In figure 7, it is clear that Cos-TFIDF and Cos-TFIDF* have different curves; similarly Cos-TF and Cos-TF* also behaves differently. In other words, cosine's behavior is different for above two cases, whereas it should be same for both cases.

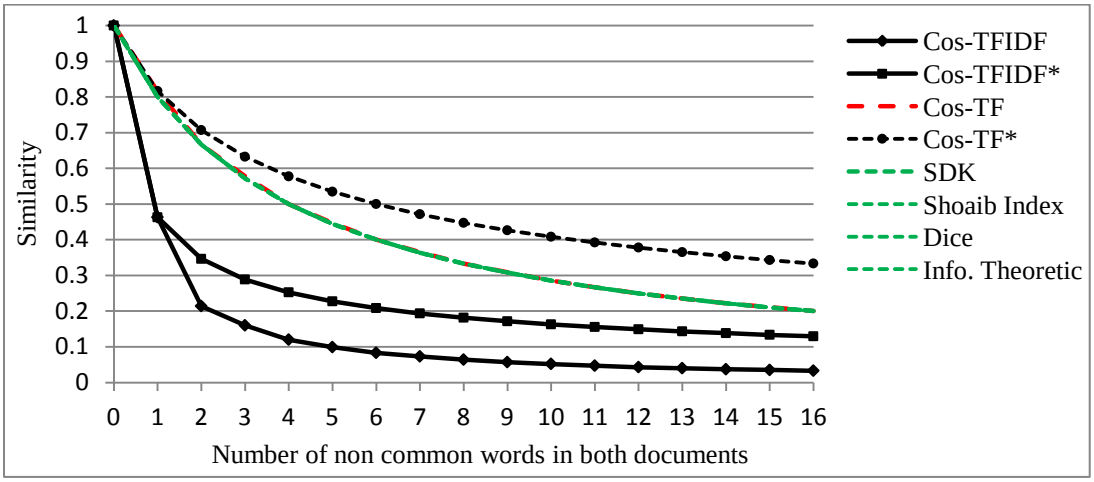

Figure 7: Effect of existence of non common words either in both documents or only in one document

SDK index, Shoaib index, Dice and information theoretic show same behavior in both cases that is why they are shown once. Cos-TF is equal to SDK index when number of non common words is same in both documents, and it is slightly higher than SDK index when one document has more number of non common words than the other.

In the above discussion we have shown that SDK index is closer to assumptions in all scenarios than baseline methods. We can conclude that SDK index is more suitable to find document similarity in pair wise fashion than baseline measures as it is the closest to the absolute measure.

3.5. Chapter Summary

We propose similarity measures for comparing citations in a pair wise fashion. Each proposed similarity measure is for different types of data. Equation 7 estimates similarity between two names; equation 9 between co-authors of two citations; equation 12 (CAM) between two titles or venues, and SDK index (equation 14) between two ref-titles of two publications. Equation 7 and 9 satisfy our assumptions I, II and III. Equation 12 is the absolute measure for documents where the term frequency of each word remains 1. It is closer to absolute value and less time consuming than cosine. SDK index resolves three main flaws of VSM based cosine. Our proposed measures can be applied to any type of textual data where name entities (not necessarily human names) or language words or both exist.

SDK index, in some cases, is equal to Shoaib index, Dice and information theoretic; and, in some cases it is better than these measures. We have compared the behavior of six well known similarity measures. Out of these, cosine measure is the farthest from absolute similarity value and SDK index is the nearest. Cosine shows reverse trend in certain conditions while SDK index doesn't. SDK index also needs not any information about the number of documents in the collection as it is needed in many VSM based similarity functions. Trying to output near to absolute value may help us decide the threshold value in clustering documents, author name disambiguation and in many other text mining tasks. Devising similarity measures with the same concept for all types of attributes in a dataset may help select a single threshold value for all types of attributes. SDK index is basically designed for textual documents. It can also be applied for entity names too.

Chapter 4. Role of References in Similarity Estimation of Publications

Chapter 4

Role of References in Similarity Estimation of Publications

In this chapter we investigate whether references play any role in similarity estimation of publications or not. We perform experiments on publication datasets to empirically confirm our notion that references are a good source of information. Experiments prove that our notion was true. AND approaches, especially unsupervised ones depend upon the similarity values of publications. Two publications are grouped into a single cluster if they have similarity value greater than threshold supposing that they are authored by the same person. The results given in this chapter prove that references are a reliable source of information.

4.1. Introduction

Similarity estimation among the publications is very important in classification and clustering techniques for grouping, indexing, citation matching and author name disambiguation purposes. Many techniques are employed in similarity estimation of publications. Publication attributes are basic source of information and play important role in similarity estimation. Most of the works in author name disambiguation like Han et al [29] use *triplet attributes.* AND works exploit diverse types of attributes such as self-citation [4] [43], abstract [4] [43], user feedback [42] topic of the publication [42] [14], author affiliation [43], authors email addresses [43] to improve similarity among publications.

In the field of academic document clustering, the use of reference markers has been of great interest to researchers. The use of reference markers and the contexts surrounding them, referred to as reference markers contexts, have also gained much attention [114] [115]. In works of Mercer et al [114] and Nanba et al [115] text surrounding a reference

marker is extracted to determine the relatedness between the two publications connected by that reference marker. Aljaber et al [116] use reference marker contexts to optimize similarity among publications.

Exploiting triplet attributes is very common practice [6] in citation matching techniques. The reason is that these attributes are available in all bibliographic databases (BDs). Using title and venue attributes to estimate publications similarity may or may not be real picture of their similarities. Two publications having totally different titles or venues may belong to the same topic(s), and on the other hand, two publications having high title or venue similarity may belong to two different areas as the title and venue attributes face scarcity of words problem. The words scarcity problem, in case of title attribute, means that a title has only few words to represent the topic(s) of a publication. In the context of venue, words scarcity means that a venue is restricted to use only few words to represent the research area(s) of that venue. So the title and venue of a publication may not be their best representatives.

It can be argued that the words scarcity problem may be resolved by comparing the complete scripts of publications. But this solution will be too much time consuming due to teh very large amount of content and hence not scalable. Another issue with complete script is that all the publications in a dataset may not be freely or easily available. On the other hand, the references are freely and easily available from almost all BDs. To overcome the words scarcity problem, Shoaib et al [105] and Shoaib et al [23] propose to utilize references (ref-titles) of the publications.

This chapter investigates the importance of references (ref-titles and ref-coauthors) in similarity estimation of publications. It is investigated that references attributes provide the similarity value that is relatively closer to the actual value. The term *actual value means the similarity calculated by comparing complete scripts of publications*. The cosine similarity function has been used to estimate title and ref-titles similarity representing the publications in the vector space model (VSM) [37].

Cosine similarity is not a better solution for matching names and estimating co-authors and ref-coauthors similarity values. The reason is that cosine matches the terms (parts of names) blindly. It is common point that a name can be written in many styles in the

citations of an author. For example, "Muhammad Shoaib Kamboh" can be written as "M. S. Kamboh", "Kamboh, M. S.", "M. Shoaib Kamboh", etc. A special methodology has been formulated to handle this problem in Shoaib et al [105] (equation 7, chapter 5). This method considers all these variant forms of name as single name. To estimate the co-authors and ref-coauthors similarity we employ equation 7 (chapter 5) along with a variant form of Jaccard coefficient given in equation 15 in this chapter.

4.2. Problem Definition

In this chapter we investigate whether ref-titles and ref-coauthors are reliable source of information for estimating publications similarity or not. We formulize the problem as:

Given two citations c_i, and c_j and their triplet attributes, investigate whether $Sim_{ref\text{-}titles}(c_i, c_j)$ and $Sim_{ref\text{-}coauthors}(c_i, c_j)$ are closer to $Sim(p_i, p_j)$ than $Sim_{title}(c_i, c_j)$, $Sim_{venue}(c_i, c_j)$ and $Sim_{co\text{-}authors}(c_i, c_j)$.

Where Sim means similarity, $Sim(p_i, p_j)$ means the actual similarity between p_i, and p_j, and subscript of Sim is the title of citation or publication.

4.3. Proposed Solution

Two different similarity measures have been applied to different types of attributes. For title, ref-titles and complete script, the state of the art cosine similarity measure has been used and the publications have been represented in VSM [37]. For author names similarity we have employed similarity measure proposed in Shoaib et al [105] specially designed for human names. For co-authors we use Jaccard like formula.

4.3.1. Similarity Measure for Title, Ref-titles and Complete Script

Cosine function based on VSM is a relative similarity measure. Shoaib et al [105] and Shoaib et al [106] propose Shoaib index and SDK index respectively. These are similarity measures for text documents that output almost absolute similarity value between two documents. Here we need not absolute values rather we need just comparisons. So, we use cosine similarity as it is the most popular measure [106] [104] for estimating document similarity based on VSM. The similarity between two publications p_i and p_j

can be defined as the normalized inner product of the two corresponding vectors $\mathbf{p_i}$ and $\mathbf{p_j}$ as given in equation 2 chapter 3. We rewrite this equation here.

$$Sim\,(p_i,\ p_j) = \frac{\mathbf{p_i}.\mathbf{p_j}}{|\mathbf{p_i}|\mathrm{X}|\mathbf{p_j}|} = \frac{\sum_{t_x \epsilon (p_i \cap p_j)} (w_{t_x}(p_i)\ x\ w_{t_x}(p_j))}{\sqrt{\sum_{t_x \epsilon p_i} w_{t_x}^2(p_i)\ x \sum_{t_x \epsilon p_j} w_{t_x}^2(p_j)}} \quad (2)$$

Where all symbols are same as in chapter 3 except p_i and p_j. Here p_i and p_j stand for i^{th} and j^{th} publications of $P = \{p_1, p_2, \ldots, p_z\}$, z is the number of publications or citations.

4.3.2. Similarity Measure for Author Names, Co-authors and Ref-coauthors

Cosine function can be applied to co-authors attribute where variations in names are minimal. It is not a better solution for entity names where a name has variant forms, especially when a name has multiple tokens (parts). The cosine function considers each variant form of a token as different term. To estimate similarity between two names $ň_i$ and $ň_j$ we exploit the similarity measure proposed by us in Shoaib et al [105]. This is given in chapter 3. We rewrite here for easiness.

$$Sim_{nam}\,(ň_i, ň_j) = \frac{(e\ *\ \alpha\ +\ b\ *\ \beta\ +q\ *\ \gamma)}{(ž\ *\ 0.5\ +\ h\ *100)} * log(ž + 2) \quad (7)$$

To estimate co-authors and ref-coauthors similarity of two publications we exploit simple jaccard like formula given in equation 15.

$$Sim_{CA}\,(p_i, p_j) = \frac{2 * (\eta_{\mathrm{thr}})}{\eta} \quad (15)$$

Where η is the total number of names in (p_i, p_j) pair; and η_{thr} is the number of names having Sim_{nam} > threshold. Sim_{nam} is estimated through equation 7. Equation 15 gives co-authors and ref-coauthors similarity between the two publications p_i and p_j.

It is to be noted that in this chapter, we are not using all of our own proposed improved similarity measures defined in chapter 3. Actually, here we are concerned to investigate whether ref-titles and ref-coauthors of two publications are reliable sources of information or not. Any similarity measure could be used here.

4.4. Results and Discussion

In this section we explain the results generated on real life publication datasets. We performed experiments on two types of datasets: i.e., publication datasets of ambiguous authors, and publication datasets of different subjects. We collected six publication datasets of different ambiguous authors as exploited by different works like [9] [29] [22]. We included only those ambiguous names and individual authors for whose publications we could collect the references along with other citation attributes. We ignored those papers for which we could not gather references either due to unavailability of data or due to cost of time to be spent to collect and prepare the references attributes. In our experimental datasets each ambiguous dataset contains 44-150 records and 3-6 individual authors. For example dataset of Ajay Gupta consists of 134 records belonging to six different Ajay Guptas. Table 12 shows statistics of six datasets. We removed stop words and performed stemming as preprocessing steps for title and ref-titles attributes.

Each ambiguous dataset has been divided into sub-datasets in such a way that each sub-dataset contains records of one and only one individual author. This step resulted into twenty eight sub-datasets.

We estimate intra sub-dataset pair wise attribute similarity for all sub-datasets. The main focus is to analyze whether references attributes help improve author name disambiguation process or not. The results are given in table 13 and table 14.

Table 13 shows a comparison between similarity values of the title and ref-titles attributes. The second column, i.e., "Intra Sub-datasets Avg. Title Sim" reports the average title similarity between the records of a sub-dataset excluding the self-comparisons. By the term self-comparison, we mean comparison of a publication with itself. Similarity value for self-comparison is always "1". Forth column reports the same thing for ref-titles attribute. Columns three and five show the time consumed in seconds to estimate corresponding attribute similarity values for intra sub-dataset records.

Table 13 shows that ref-titles similarity is always greater than title similarity. For example, ref-titles similarity is 2.48 times greater than title similarity for Rakesh Kumar dataset, and 1.31 times for Cheng Chang dataset. On the average ref-titles similarity is 1.7 times greater than title similarity. Estimating ref-titles similarity is comparatively more

time consuming than estimating title similarity. On the average time consumed to calculate ref-titles similarity is 1.8 times greater than the time consumed for title similarity. The disadvantage of greater time consumption is negligible as compared to the advantage of similarity information from ref-titles attribute. Table 13 shows that ref-titles similarity is the more reliable source of information for publications datasets of ambiguous authors.

Table 14 is similar to table 13 with the only difference that it shows similarity values for co-authors and ref-coauthors attributes.

Table 14 shows that for some datasets (e.g., Ajay Gupta) ref-coauthors similarity is greater than co-authors similarity, and for some datasets (e.g., Jim Smith) situation is reversed. For example, ref-coauthors similarity is 1.56 times of co-authors similarity for Ajay Gupta dataset, and 0.41 times for Jim Smith's dataset. On the average co-authors similarity is 1.45 times greater than ref-coauthors similarity. Estimating ref-coauthors similarity is more time consuming than estimating co-authors similarity. On the average time consumed to calculate ref-coauthors similarity is 3.88 times greater than the time consumed for co-authors similarity. The disadvantage of additional time consumption is bearable. In the trade of CPU time cost, we get an additional source of information for publications dataset. Table 14 reveals that although ref-coauthors attribute is not as powerful source of information as co-authors attribute yet it is a useful source of information for publications datasets of ambiguous authors and it should be used as an additional attribute of publications in AND process.

Now let us analyze whether ref-titles and ref-coauthors similarity is closer to actual similarity than title, co-authors and venue similarity or not. We have prepared three small datasets of 30 publications each from different subject. Each tiny dataset contains publications on of the same topic from respective subject. These datasets are not from the same author or the same ambiguous name instead they are from the same topic. For these datasets, title, ref-titles, co-authors, ref-coauthors, venues and complete script similarities have been estimated. The results are shown in table 15. Similarity values for title, ref-titles, venues and complete scripts mentioned in table 15 are estimated through VSM

based cosine equation 2 (chapter 3); for names equation 7 (chapter 3) is utilized; and for co-authors and ref-coauthors equation 15 is exploited.

Table 15 shows ref-titles similarity is the closet to actual similarity (complete script similarity), and it is almost 3 times higher than title similarity. It is concluded that ref-titles are good source of information for topic based publications datasets.

Table 12: Publication datasets of ambiguous authors

Ambiguous Names	No. of Records	No. of Authors	Ambiguous Names	No. of Records	No. of Authors
Ajay Gupta	134	6	Hui Fang	87	4
Bing Liu	105	5	Jim Smith	44	3
Cheng Chang	61	4	Rakesh Kumar	150	6

Table 13: Comparison between similarity values of title and ref-titles attributes

Ambiguous Name	Intra Sub-datasets Avg. Title Sim	Time Consumed (sec.)	Intra Sub-datasets Avg. Ref-titles Sim	Time Consumed (sec.)
Ajay Gupta	0.033946824	0.7644013	0.061333268	1.3572023
Bing Liu	0.024543994	0.670203	0.058223043	1.2932041
Cheng Chang	0.060755893	0.6096011	0.079818487	0.9204017
Hui Fang	0.044190486	0.6396011	0.076130073	1.1204016
Jim Smith	0.055144755	0.4212007	0.07249837	0.8112015
Rakesh Kumar	0.025779532	0.9360016	0.063823024	1.8096032
Total	0.244361485	4.0410088	0.411826264	7.3120144

Table 14: Comparison between similarity values of co-authors and ref-coauthors attributes

Ambiguous Name	Intra Sub-datasets Avg. Co-auths Sim.	Time Consumed (sec.)	Intra Sub-datasets Avg. Ref-coauths Sim.	Time Consumed (sec.)
Ajay Gupta	0.154066219	1.1870165	0.240939995	6.2480121
Bing Liu	0.117684718	1.2840735	0.180463357	9.5620231
Cheng Chang	0.464357143	1.4570832	0.302015341	1.6700025
Hui Fang	0.317898957	1.5990917	0.175691372	3.2340073
Jim Smith	0.420530456	1.3640782	0.174297317	4.4920047
Rakesh Kumar	0.329393691	1.5520035	0.173592064	8.5570157
Tot.	1.803931184	8.4433466	1.246999445	32.7630654

Table 15: Comparison between similarity values of title, ref-titles, co-authors, ref-coauthors and venue attributes w.r.t. actual similarity (complete script sim)

Datasets	Title Sim	Ref-titles Sim	Co-authors Sim	Ref-coauthors Sim	Venue Sim	Complete Script Sim
Computer Sc.	0.046	0.111	0.015	0.027	0.017	0.304
Physics	0.028	0.160	0.013	0.012	0.002	0.155
Economics	0.031	0.052	0.004	0.002	0.001	0.121
Total Sim	0.105	0.323	0.032	0.041	0.020	0.580
Average Sim	0.035	0.108	0.011	0.014	0.007	0.19

While analyzing references attributes of publications, we get some interesting pieces of information.

- Consider two publications of Rakesh Kumar[36] having the same title and co-authors published in two different venues. Their title and co-authors similarity is 1.00, and venue similarity is 0.034. Title and co-authors similarity reveal that these two publications are not different publications while venue similarity depicts that they are two different publications. Ref-titles and ref-coauthors similarity values (0.280877 and 0.779 respectively) show that they share a reasonable amount of data. To estimate real similarity picture between the two publications we compared their abstracts and then complete scripts. Abstract and complete script similarity values are 0.348094 and 0.544173 respectively. Out of all these values full script similarity value, i.e., 0.544173 is the most reliable and genuine, and we name it actual similarity value. The similarity value of ref-titles has the least deviation from actual value (i.e., |0.544173 - 0.779| = 0.234827).
- Consider two publications[37] of Ajay Gupta having the same title and co-authors but different venues. Their title and co-authors similarity is 1.00, and venue similarity is 0.073. Title and co-authors similarity reveal that these two publications are not different publications while venue similarity depicts that they may share little amount of data. Ref-titles and ref-coauthors similarity values (0.912 and 1.00 respectively) show that they share almost whole text. To estimate real similarity picture between

[36] R. Kumar, Y. Shan, H. S. Sawhney, "Unsupervised Learning of Discriminative Edge Measures for Vehicle Matching between Non-overlapping Cameras". The first paper was published in *Computer Vision and Pattern Recognition,* and second in *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
[37] A. Gupta and B. Beckmann, "PANSY: A Portable Autonomous Irrigation System". The first paper was published in *Journal of Indian Society of Agricultural Statistics,* and second in *International Conference on Statistics and Informatics in Agricultural Research.*

the two publications we compared their complete scripts. Complete script similarity value was also equal to 1.0. Out of all these values full script similarity value, i.e., 1.0 is the most reliable and genuine. In this case the similarity values of all attributes except venue are very close or equal to the actual value i.e., 1.0. After getting high similarity value of ref-titles and ref-coauthors, we manually investigated the two publications. Our notion was that they would be the exact copy of each other. After investigation, it proved that our notion was absolutely true.

From above discussion, it is concluded that if two publications have the same titles and co-authors but vary in their references, then one of them may be the extension of the other. Also, if two publications have the same titles, co-authors and references, then it is quite possible that they are copies of each other. From this discussion, it reveals that references attributes help a large in certain situations to decide whether two documents are copies of each other or not. This simple test may also be performed to help decide the plagiarism process.

The purpose is to investigate whether references attributes (ref-titles and ref-coauthors) could be used as sources of information to estimate academic document similarity. We started this research with two notions: (1) authors include those references which relate to the topic(s) of the publication; (2) ref-coauthors would be good source of information for estimating publications similarity. Both of our notions are true as they have been proved by empirical results generated from real life datasets. Above results and discussion show that references (ref-titles and ref-coauthors) are a useful source of information for publication datasets in the process of author name disambiguation.

References attributes help improve text mining tasks like clustering and classification by improving document similarity. Document clustering and AND process mainly base on documents similarity. References attributes provide good source of similarity information for academic documents. Resultantly they will surely improve academic document clustering, AND process, and many other tasks which rely on document similarity. References attributes provide reliable information about the amount of data the two documents share with each other. Ref-titles attribute is more reliable as compared to the title attribute. Ref-coauthors attribute though not more informative than co-authors

attribute yet provides a reasonable amount of similarity information. From this discussion, we conclude that our proposed idea of exploiting references for estimating academic document similarity is worthwhile.

4.5. Chapter Summary

In this chapter, it has been proven that references attribute (ref-titles and ref-coauthors) help improve publications similarity. Experiments have been performed on publication datasets of ambiguous authors, and publication datasets having same topic. Text mining tasks like author name disambiguation and academic document clustering base mainly on publications similarity. From experiments, it is concluded that references attributes provide a good source of similarity information for publications. Ref-titles attribute is more reliable as compared to the title attribute. Ref-coauthors attribute, though not more informative than co-authors attribute yet provides a reasonable amount of similarity information. From this discussion, it is concluded that the proposed idea of exploiting references for estimating academic document similarity is worthwhile.

Chapter 5. Author Name Disambiguation

Chapter 5

Author Name Disambiguation

In this chapter we propose a solution for author name disambiguation in bibliographic databases. We employ our own proposed clustering algorithm that is modified form of clustering algorithm proposed by Cota et al [25] and Ferreira et al [22]. We focus to utilize improved similarity measures proposed by us and explained in chapter 3.

5.1. Introduction

In digital libraries (DLs) and bibliographic databases (BDs) it has been observed that multiple authors share a common name or a single author may appear with different names. This sharing or variation causes author name ambiguity in DLs and BDs. Resolving author name ambiguity in citations of a BD is referred as author name disambiguation (AND).

The majority of works (almost all) has focused performance (accuracy and scalability) a lot, but following aspects of AND have been least focused.

- Selecting the appropriate seeds for clusters
- Predicting actual authors of new citations while populating the BD with new records.
- Exploiting appropriate similarity measures for publication attributes
- References, a good source of information have not been utilized

An important issue of clustering citations in AND is the selection of appropriate seeds. It is essential for better performance in the clustering process. Initial entries of clusters, especially the seed play important role in clustering process. Initial few wrong entries, especially in hierarchical clustering, may affect performance adversely. That is why we focus on the selection of appropriate seed. Another issue faced by AND is prior knowledge about the number of actual authors. Unfortunately this is very hard to get this

information, and it is not available at all for new citations in real scenarios. Cosine similarity in vector space model [37] is the most used similarity measure [106] [104]. In chapter 3 we have proved that cosine measure shows inverse trend in some scenarios. That is why we prefer to use similarity measure proposed by Shoaib et al [105] and Shoaib et al [106] specifically designed for AND problem.

References as are a reliable source of information [23]. Chapter 4 gives summary of the similarity values of title, co-authors, ref-titles and ref-coauthors attributes. We have also employed ref-titles attribute as additional sources of information in our clustering algorithm approach.

5.2. Problem Definition

Here, we address the author name disambiguation problem defined in chapter 2. We rewrite the problem for readers' convenience.

Given a set of citations $C = \{c_1, c_2, \ldots, c_z\}$ *sharing same author name A, group C into k disjoint clusters* $G = \{g_1, g_2, \ldots g_k\}$ $(1 \leq k \leq z)$ *such that citations within each cluster* g_i *belong to the same author* $a_i \in A$*, and no citation* c_i *is member of any two clusters, i.e.,* $g_i \cap g_j = \phi$.

5.3. Proposed Solution

In this section we explain our proposed author name disambiguation methodology.

5.3.1. Blocking Step

To disambiguate citations of a BD, we split the citations into ambiguous groups. These groups can be obtained by using any blocking method [25] [70]. Blocking methods improve performance by avoiding unnecessary comparisons among the citations of non compatible authors.

Works like [25] [69] [70] [22] utilize FCM like methods to obtain ambiguous groups by matching first initial and last name only. FCM may cause non matching names to join the same ambiguous group. For example, according to FCM, "A. K. Gupta" and "A. B. Gupta" both are combined to "A. Gupta" ambiguous group, whereas they are two different names representing two different authors. To avoid such scenarios we propose a slightly different approach than existing name blocking methods used in AND.

Ambiguous groups of polysemy names can easily be obtained as all the name occurrences belonging to an ambiguous group are exactly same (dentical). On the other hand, ambiguous groups of synonym names require much more attention. It is more critical whether a name variant should be included in to an ambiguous group or not.

We use name similarity estimated through equation 7 (chapter 3) and insert a citation to an ambiguous group only if its name similarity value exceeds threshold. In other words, we combine only common names (see chapter 3) in an ambiguous group. Equation 7 outputs Sim (A. K. Gupta, A. B. Gupta) near to zero, and hence there are no chances for such type of wrong entries. The selection of threshold needs much care; if it is too high it may restrict relevant entries, and on the other hand, if it is too low then it may cause wrong citations to join an ambiguous group.

5.3.2. Name Similarity

The name similarity between two names $ň_i$ and $ň_j$ is useful in blocking step as well as in estimating co-authors similarity. Similarity between two names $ň_i$ and $ň_j$ can be estimated through any similarity function, but we use equation 7 (chapter 3) because it is closer to our assumptions I and II (chapter 3) than existing. For ready reference we rewrite that equation here.

$$Sim_{nam}\,(ň_i, ň_j) = \frac{(e * \alpha + b * \beta + q * \gamma)}{(ž * 0.5 + h * 100)} * log(ž + 2) \qquad (7)$$

This equation is capable to assign different weights to different types of tokens. Why do we assign different weights to different types of tokens? It is explained in chapter 3.

5.3.3. Co-authors Similarity

Any similarity function like Jaccard, Dice, information theoretic, etc. can be used to estimate co-authors similarity, but all these methods estimate similarity that is proportional to the common and total co-author names. These methods do not satisfy our assumption III (chapter 3). To depict assumption III in co-authors similarity we employ equation 9, chapter 3 along with equation 7 proposed by Shoaib et al [105].

$$\mathrm{Sim}_{\mathrm{CA}}(c_i, c_j) = lo\,g\left(\sum\left(Sim_{nam}(\check{\mathrm{n}}_x\,,\,\check{\mathrm{n}}_y)\right)+1\right) - \frac{\log(\eta'+1)}{\log(\eta+1)} * \frac{1}{10+\eta+1} \qquad (9)$$

5.3.4. Title and Venue Similarity

For title and venue attribute we use CAM (equation 12, chapter 3) proposed by Shoaib et al [105] because its output is the absolute value for texts where term frequency is 1. We observe that titles and venues of publications usually don't have any repeating word.

$$Sim_{title}(c_i, c_j) = \frac{(|tw_{c_i}| \cap |tw_{c_j}|)}{(|tw_{c_i}| \cap |tw_{c_j}|) + ((|tw_{c_i}| \cup |tw_{c_j}|) - (|tw_{c_i}| \cap |tw_{c_j}|)) * 0.5} \qquad (12)$$

All symbols of the above equation are explained in chapter 3. Why do we employ CAM instead of cosine in the vector space model [37] frequently used in literature? The answer of this question is provided in chapter 3.

5.3.5. Ref-titles Similarity

A citation has a few words in its title and venue. This scarcity of words may cause, in some cases, the inappropriate similarity value. Shoaib et al [23] propose to utilize titles of references (ref-titles) of publications as additional source of information. We combine all titles of references of a publication into one title and name it as ref-titles (references titles). If we have *r* references of a publication *p* then there are *r* titles as each reference has exactly one title. Concatenation of *r* titles gives us *ref-titles* attribute. CAM (equation 12) is a better choice for citation titles and citation venues where a word hardly repeated. In ref-titles, a word may repeat itself many times, and CAM and other existing measures are not a better solution for such situations.

Shoaib et al [106] propose SDK index (chapter 3) to estimate the ref-titles similarity, and prove that this index is a better choice than cosine measure and many other existing ones. Shoaib et al [23] exploit ref-titles and prove that ref-titles are a good source of information. We propose using SDK index for ref-titles attribute (*rt*) and it is rewritten here for convenience.

$$\mathrm{Sim}_{rt}(p_i, p_j) = \frac{\sum_{t_x \epsilon u}\left(\frac{1}{1+\log\left(\max\left(f_{t_x}(p_i,p_j)\right)/\min\left(f_{t_x}(p_i,p_j)\right)\right)}\right)}{u + 0.5 * u' + \sum_{t_x \epsilon u}(log\ (maxf_{t_x}(p_i,p_j) - minf_{t_x}(p_i,p_j)))^2 + (\sum_{t_y \epsilon u'} log\ (f_{t_y}(p_i,p_j))} \quad (14)$$

Where all symbols are same as they are in equation 14 (chapter 3).

5.3.6. Seed-based Hierarchical Clustering

In this section we provide description of SHC algorithm, its pseudo code and its complexity. First of all we enlist notations used in SHC algorithm in table 16.

Table 16: Mathematical notations used in following algorithm

Symbols	Sets	Description
A	A= {a_1, a_2, …, a_k} where a_i is the *ith* author; k is # of unique authors sharing an ambiguous name	Set of authors/persons sharing an ambiguous name *A*.
P	P= {p_1, p_2, …, p_z} z is number of publications belonging to A	Set of publications associated to an ambiguous name
C	C = {c_1, c_2, …, c_z} z is number of citations belonging to A	Set of citations associated to an ambiguous name
sk_i	Seed of cluster k_i	
C' M M'	C'= C-M, M= {c_1, c_2, …, c_b} M'= M - {{sk_i} U {ssk_i}}, ssk_i = citations that are same as sk_i (i.e., citations that have Sim_{CA} with sk_i close to 1) Each c_i is a member of any cluster k_i. b is the number of citations that have been included in clusters.	
C		Vector form of citation in VSM
Note: P and C are different only in a sense that former denotes the complete publication, whereas the later denotes citation of the publication.		
K $Ƙ$ $Ƙ'$ $\check{K}$	K={k_1, k_2, …,k_q} $Ƙ$= {k_1, k_2, …,k_s}= set of clusters arranged in descending order of size; k_1 being the largest and first, and k_s being the smallest and the last cluster. (There may be multiple largest and smallest clusters) $Ƙ'$= {k_2, k_3, …,k_s} $\check{K}$ = {k_1, k_3…,k_{s-1}} Different sets of clusters; q is the number of clusters predicted as associated to an ambiguous name *A*. Ideally q should be equal to k.	No. of clusters constructed against an ambiguous name *A*.
Sim_{CA}	Co-authors similarity	
Sim_{CA} (sk_i, c_i)	Co-authors similarity between sk_i and c_i	

5.3.6.1. Seed-based Hierarchical Clustering Algorithm

Input: $C = \{c_1, c_2, \ldots, c_z\}$, citation dataset.

Output: $K=\{k_1, k_2, \ldots, k_q\}$, ideally q should be equal to the number of actual authors belonging to A.

Begin

Phase I

Step 1. While (C !=null)

Step 2. Create a new cluster k_i

Step 3. For each citation $c_i \in C'$ calculate cumulative Sim_{CA} with all citations of C' using equations 7 and 9.

Step 4. Select a citation c_i which has MAX value of cumulative Sim_{CA} as seed of cluster k_i and remove it from C'. $c_i \in C'$ and $k_i \in K$.

Step 5. Successively combine each c_i to k_i and remove it from C' and insert to k_i IF $\text{Sim}_{CA}(sk_i, c_i) >$ threshold; where $c_i \in C'$

Step 6. Repeat step 5 for each $c_{m'}$; where $c_{m'} \in M'$

Step 7. End While

Phase II

Step 8. Arrange all clusters created in phase I in descending order of size. This gives set $\vec{K}$.

Step 9. For (i= index of k_s; $i > 0$; i--) //Loop for $\vec{K}$ starting from k_s and ending at k_2. Suppose index starts from 0.

Step 10. While (continue = =true)

Step 11. For each citation $c_m \in k_i$

Step 12. For (l=0; l<index of k_i; l++) //Loop for $\check{K}$ starting from k_1 and ends at k_{s-1}

Step 13. For (n=0; $n <$ size of k_l; n++)

Step 14. Compare similarity of all available attributes (except co-authors) between c_m and c_n (and between their corresponding publications i.e., p_m and p_n). where $c_m \in k_i$ and $c_n \in k_l$.

Step 15. IF votes for combine YES are more than or equal to combine NO
THEN combine cluster k_i to k_l && continue = false; //&& is logical operator
End IF

END Inner For Loop

END Middle For Loop

END Foreach Loop

END While Loop

END Outer For Loop

Step 16. End

5.3.6.2. Description of SHC Algorithm

Our proposed SHC algorithm performs AND process in two phases. In phase I, it uncovers only co-authors patterns among citations and tries to construct clusters. It selects the most appropriate citation as seed of a cluster k_i by estimating cumulative co-authors similarity of all citations belonging to *C*. Co-authors similarity of each citation c_i of *C* is calculated against all citations of *C* by exploiting equations 7 and 9. In this way we get a co-authors similarity matrix of *C*X*C*. We sum up each column of this matrix individually and select one that shows maximum cumulative co-authors similarity as the seed of k_i. This seed (citation) is removed from *C*. Each citation c_i of *C'* is compared to the seed of k_i (sk_i) and combined to the cluster k_i if $\text{Sim}_{\text{CA}}(sk_i, c_i) >$ threshold. Each citation that is combined to cluster k_i is removed from the citation list and added to cluster k_i and we get *C'* = *C*-*M*. We further populate cluster k_i by successively comparing each element of *M'* to the citations list *C'*. This successfully resolves the problem of transitive co-authors if exit in citations dataset. After cluster k_i stops being populated and if *C'* is not empty, then successively a seed for new cluster (say, k_{i+1}) is estimated and above process is repeated for new cluster. At the end of phase I we have all citations in different clusters and no citation in *C* or *C'*. Value of threshold plays important role in the clustering process. If we lower the threshold value, phase I will create small number of clusters as compared to the actual number of clusters (authors) causing low precision and recall. On the other hand, if we set the high threshold value, it will produce a large number of clusters cause fragmentation. Fragmentation means citations of the same person exist in two or more clusters.

Phase II uses as many attributes as are available except co-authors attribute. We use title, venue and ref-titles attributes. We estimate title and venue similarity through CAM (equation 12, chapter 3), ref-titles similarity through SDK index (equation 14, chapter 3), and. We employ voting scheme that helps decide whether to combine two clusters or not.

The clusters are combined if a majority of attributes favors to combine. We brief this process here. We arrange all clusters produced in phase I in descending order of size and get set $Ƒ$. We try to combine each cluster k_i of $Ƒ'$ (starting from k_s and ending at k_2) to any of the cluster k_j of $Ǩ$ (starting from k_j and ending at k_{s-1}). Unlike works in [25] [22], we neither aggregate the information of a cluster nor compare clusters in a pair wise fashion. We compare their citations in pair wise fashion estimating the similarity of all available attributes except co-authors attribute. If an attribute's similarity is greater than threshold, it is considered a "combine YES" vote, otherwise it is a "combine NO" vote. If number of "combine YES" votes is greater than that of "combine NO" ones, the respective clusters are fused (combined) into one. In other words a cluster k_i of $Ƒ'$ is fused to a cluster k_j of $Ǩ$ if and only if k_i gets majority votes to be combined to k_j. In case of tie we combine the respective clusters too.

The above algorithm is for only one ambiguous author name dataset. We repeat this algorithm as many times as number of ambiguous author datasets. Our proposed algorithm is unsupervised and it requires no human efforts like labeling the citations to their respective actual authors. It is capable of predicting the actual number of authors by producing the number of clusters equal (or close) to the actual number of authors sharing an ambiguous name *A*. It successfully distributes citations in produced clusters in such a way that no citation is the member of two or more clusters and each cluster has high precision and recall. It selects appropriate seeds for each cluster. It is also helpful while incremental updates of the bibliographic database. It can easily either detect the actual author of a new citation if his/her citations are already present in BD or predict that new record (citation) belongs to the new author.

5.3.6.3. Complexity of SHC Algorithm

Complexity of Phase I

We have z number of citations and q number of clusters constructed by algorithm 1. Suppose that every cluster has an equal number of citations (i.e., z/q citations).

Step 2 runs exactly q times so its complexity is $O(q)$

The complexity for finding seed (step 3) is $O(z^2/q^2)$.

The complexity of step 5 is $O(z/q)$.

The complexity of step 6 is also $O(z^2/q^2)$.

The complexity of step 3-6 is = $O(z^2/q^2)$+ O (z) + $O(z^2/q^2)$ = $[O(z^2/q^2)]$. By the property of O.

The complexity of phase I will be $O(q)$ $O(z^2/q^2)$ = $O(z^2/q)$ By property of O.

Complexity of Phase II

The outer most loop (step 9) will run exactly q times.

The loop in step 11 will also run exactly (on average) z/q times.

The loop in step 12 will run exactly (on average) $q/2$ times.

The inner most loop (step 13) will run exactly (on average) z/q times.

For finding attributes similarity there is 1 comparison for each attribute. So there are only 4 comparisons as there are 4 attributes to be compared.

The overall complexity of phase II can be written as:

q * (z/q * ($q/2$ * (z/q * (4)))) = 2(z^2). This can be written as O (z^2) by the property of O.

Complexity of Phase I and II

Combining the complexities of phase I and II we get:

Complexity of algorithm 1 = $O(z^2)$+ $O(z^2/q)$ = $O(z^2)$ by property of O.

This complexity is same as those of the algorithms of Cota et al [25] and Ferreira et al [22]. Our approach gives better performance than baselines with the same complexity.

Our approach resolves information scarcity problem by exploiting ref-titles and ref-coauthors attributes. In phase II, it also resolves fragmentation problem produced in phase I along with exploiting other available information sources.

5.4. Experimental Setup

In this section we explain the experimental setup and the results generated on three different types of collections. We implemented the AND process in C#.Net. We perform stemming and stop words removal as a preprocessing step for all datasets.

5.4.1. Datasets

We perform experiments on three different types of collections. First and second types of collections consist of BDBComp and DBLP publications. These are same as used by Ferreira et al [22] and Cota et al [25]. We downloaded these collections from http://clgiles.ist.psu.edu/data/. The statistics of these collections are reported in Table 17. These collections contain co-author, title and venue attributes only.

Table 17: DBLP and BDBComp Publication Collections.

BDBComp Collection				**DBLP Collection**			
Ambiguous Authors	**No. of Citations / No. of Authors**	**Ambiguous Authors**	**No. of Citations / No. of Authors**	**Ambiguous Authors**	**No. of Citations / No. of Authors**	**Ambiguous Authors**	**No. of Citations / No. of Authors**
A. Oliveira	52/20	J. Souza	34/12	A. Gupta	576/26	J.Robinson	171/12
A. Silva	64/38	L. Silva	33/18	A. Kumar	243/14	J. Smith	904/29
F. Silva	27/22	M. Silva	21/16	C. Chen	798/61	K. Tanaka	280/10
J. Oliveira	48/22	R. Santos	20/17	D. Johnson	368/15	M. Brown	153/13
J. Silva	35/18	R. Silva	27/22	J. Martin	11216	M. Jones	260/13
---	---	---	---	---	---	M. Miller	405/12

The Third type of collection consists of publication datasets of seven ambiguous authors from DBLP as used by Shoaib et al [23]. This collection contains ref-titles attribute along with title and venue attribute. The statistics of this collection are shown in Table 12 (chapter 4). Here we name it as DBLP-Ref collection just to distinguish the former DBLP collection.

5.4.2. Evaluation Metrics

We use precision, recall, F-measure, average clustering purity (ACP), average author purity and K-measure to evaluate our method and to compare with baseline methods.

5.4.2.1. Precision

It is the ratio between the number of correctly predicted citations of author a_i and number of citations predicted as a_i's citations. Equation 16 gives a mathematical definition of precision.

$$\boldsymbol{Precision} = \frac{\{\boldsymbol{c}_{\boldsymbol{a}_i}\} \cap \{\boldsymbol{c'}_{\boldsymbol{a}_i}\}}{\{\boldsymbol{c'}_{\boldsymbol{a}_i}\}} \tag{16}$$

Where $\{c_{a_i}\}$ is the set of citations of authora_i; and $\{c'_{a_i}\}$ is the set of citations predicted as author a_i's.

5.4.2.2. Recall

It is the ratio between the number of correctly predicted citations of author a_i and number of a_i's citations. Equation 17 gives mathematical definition of recall.

$$Recall = \frac{\{c_{a_i}\} \cap \{c'_{a_i}\}}{\{c_{a_i}\}} \tag{17}$$

In equation 17, all symbols are same as they are in equation 16.

5.4.2.3. F-measure

It is calculated using precision and recall. Equation 18 gives mathematical definition of f-measure.

$$F - measure = \frac{Precision * Recsll}{Precision + Recall} * 2 \tag{18}$$

5.4.2.4. Average Clustering Purity

Given an ambiguous author A, average clustering purity (ACP) evaluates the purity of the empirical clusters with respect to the theoretical clusters for this ambiguous author. The ACP is mathematically defined in the following equation.

$$ACP = \frac{1}{Z}\sum_{i=1}^{e}\sum_{j=1}^{t}\frac{n_{ij}^2}{n_i} \tag{19}$$

Where *z* is the total number of citations in ambiguous group (i.e., total number of citations associated with an ambiguous author name), *t* is the total number of theoretical clusters associated with ambiguous author, *e* is the number of empirical clusters for this ambiguous author, n_i is the total number of citations in the empirical cluster *i*, and n_{ij} is the number of citations in empirical cluster *i* which are also in the theoretical cluster *j*.

5.4.2.5. Average Author Purity

For a given ambiguous name *A*, average author purity (AAP) evaluates the fragmentation of the empirical clusters with respect to the theoretical clusters. AAP is mathematically defined in the following equation.

$$AAP = \frac{1}{Z}\sum_{j=1}^{t}\sum_{i=1}^{e}\frac{n_{ij}^2}{n_j} \tag{20}$$

Where n_j is the total number of citations in theoretical cluster *j*.

5.4.2.6. K-Measure

The K-measure is the geometric mean between ACP and AAP. It evaluates the purity and fragmentation of the empirical clusters. The K-metric is mathematically defined in the following equation.

$$K - measure = \sqrt{ACP * AAP} \tag{21}$$

5.4.3. Baselines

We use heuristic based hierarchical clustering (HHC) by Cota et al [25] and SAND by Ferreira et al [22] as baseline methods to compare our proposed clustering algorithm because these methods are very much similar to our proposed algorithm. Both of these baseline methods claim to find number of actual authors in an ambiguous dataset automatically. First approach employs heuristic based hierarchical clustering and the second uses automatic self-training algorithm to overcome the disambiguation process. We have explained these approaches in chapter 2, section 2.1.5.

5.5. Results and Discussion

In this section we report the results generated on three different types of collections. Most of the results are provided in tables only, and some important results are shown in graphs along with tables. The best results are highlighted in bold faced text in tables.

Where ever we use SHC that does not mean only our proposed algorithm rather it means our proposed algorithm exploiting our proposed similarity measures. If in some places SHC means only our proposed algorithm we will explicitly mention it.

5.5.1. Results against Each Evaluation Measure

These experiments include title, co-authors and venue attributes only. Ref-titles attribute is not included in these experiments as this attribute is not available in DBLP and BDBComp collections. Here we report ACP, AAP, K-measure, precision, recall and F-measure values on three different collections generated by SHC.

Table 18 reports ACP, AAP, k-measure, precision, recall and f-measure values for BDBComp datasets generated by SHC.

Table 18: Results of SHC on BDBComp datasets

Ambiguous Author Name	ACP	AAP	K-Measure	Precision	Recall	F-Measure
A. Oliveira	0.969	0.878	0.922	0.957	0.8696	0.911
A. Silva	1.0	0.940	0.969	1.0	0.927	0.962
F. Silva	1.0	0.96	0.98	1.0	0.957	0.978
J. Oliveira	0.928	0.91	0.937	0.958	0.917	0.937
J. Silva	1.0	0.797	0.893	1.0	0.783	0.878
J. Souza	0.956	0.947	0.951	0.923	0.923	0.923
L. Silva	1.0	0.939	0.97	1.0	0.900	0.947
M. Silva	1.0	0.952	0.976	1.0	0.941	0.97
R. Santos	1.0	0.950	0.975	1.0	0.944	0.971
R. Silva	0.963	0.963	0.963	0.957	0.957	0.957
Avg.	0.982	0.924	0.954	0.98	0.912	0.943

Table 19 reports values of different evaluation measures generated by SHC on DBLP datasets.

Table 19: Results of SHC on DBLP datasets

Ambiguous Author Name	ACP	AAP	K-Measure	Precision	Recall	F-Measure
A Gupta	0.662	0.99	0.809	0.586	0.897	0.709
A Kumar	0.333	0.927	0.556	0.591	0.636	0.613
C Chen	0.180	0.919	0.407	0.3376	0.645	0.475
D Johnson	0.62	0.917	0.760	0.774	0.484	0.596
J Martin	0.838	0.850	0.844	0.923	0.615	0.738
J Robinson	0.692	0.781	0.735	0.833	0.5	0.625
J Smith	0.577	0.851	0.686	0.634	0.876	0.745
K Tanaka	0.562	0.786	0.665	0.875	0.313	0.461
M Brown	0.697	0.924	0.803	0.833	0.722	0.774
M Jones	0.836	0.825	0.830	0.902	0.317	0.469
M Miller	0.835	0.993	0.911	0.857	0.857	0.857
Avg	0.621	0.888	0.728	0.741	0.624	0.642

Table 20 reports values of different evaluation measures generated by SHC on DBLP-Ref datasets.

Table 20: Results SHC on DBLP-Ref datasets

Ambiguous Author Name	ACP	AAP	K-Measure	Precision	Recall	F-Measure
A Gupta	0.966	0.949	0.957	0.857	0.571	0.686
B Liu	0.564	0.912	0.717	0.857	0.429	0.571
C Chang	1.0	1.0	1.0	1.0	1.0	1.0
H Fang	0.875	0.637	0.746	0.857	0.571	0.686
J Smith	0.896	0.836	0.866	0.8	0.6	0.686
K Zhang	1.0	0.571	0.756	1.0	0.667	0.8
R Kumar	0.976	0.808	0.888	0.933	0.333	0.491
Avg.	0.897	0.816	0.847	0.901	0.596	0.703

5.5.2. Comparison with Baselines

Table 21 and Figure 8 report the comparison for k-measure and f-measure values of HHC, SAND and SHC on BDBComp datasets. Table 21 and Figure 8 show that our method outperforms HHC by 28.8% in k-measure and 58.22% in f-measure; and SAND by 13.1% in k-measure and 57.96% in f-measure. Table 21 reports k-measure and f-measure values against each dataset of BDBComp collection for each method, whereas Figure 8 shows a comparison of SHC with baselines for average values of all BDBComp datasets.

Table 21: Comparison of SHC with baseline methods on BDBComp datasets

Ambiguous	HHC		SAND		SHC	
Author Name	K-Measure	F-Measure	K-Measure	F-Measure	K-Measure	F-Measure
A. Oliveira	0.821	0.78	0.842	0.811	**0.922**	**0.911**
A. Silva	0.801	0.584	0.814	0.678	**0.969**	**0.962**
F. Silva	0.852	0.487	0.885	0.544	**0.98**	**0.978**
J. Oliveira	0.729	0.654	0.817	0.711	**0.919**	**0.937**
J. Silva	0.656	0.628	0.856	0.674	**0.893**	**0.878**
J. Souza	0.723	0.761	0.85	0.718	**0.951**	**0.923**
L. Silva	0.755	0.604	0.851	0.658	**0.97**	**0.947**
M. Silva	0.825	0.313	0.811	0.366	**0.976**	**0.97**
R. Santos	0.822	0.621	0.942	0.571	**0.975**	**0.971**
R. Silva	0.405	0.524	0.75	0.235	**0.963**	**0.957**
Avg	0.74	0.596	0.842	0.597	**0.952**	**0.943**
Improvement	Over HHC				**28.8**	**58.221**
%	Over SAND				**13.1**	**57.957**

The results of SHC on BDBComp datasets are surprisingly higher than the baselines. Why SHC's performance is so high? The reason is only that most of the ambiguous author names in these datasets are shown as full names instead of short names. Baseline methods use either FCM or Jaccard coefficient to differentiate two names. FCM

considers two names having intermediate conflicting tokens as same. Similarity Jaccord coefficient considers two names having one conflicting token as same name. Contrary to these methods our proposed name similarity measure considers such names two different names by assigning very low similarity value. We manually investigated that all the pair of names under one ambiguous name having one conflicting token refer to two different authors (persons).

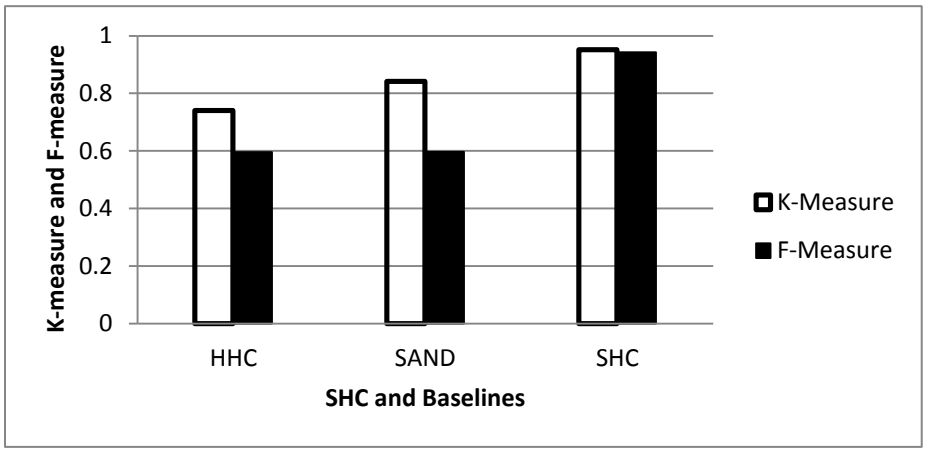

Figure 8: Comparison of SHC with baseline methods on BDBComp datasets

Table 22 and Figure 9 report the comparison against k-measure and f-measure values of HHC, SAND and SHC on DBLP datasets.

Table 22: Comparison of SHC with baseline methods on DBLP datasets

Ambiguous Author Name	**HHC**		**SAND**		**SHC**	
	K-Measure	**F-Measure**	**K-Measure**	**F-Measure**	**K-Measure**	**F-Measure**
A Gupta	0.646	0.716	0.753	0.725	**0.809**	0.709
A Kumar	0.652	0.547	0.611	0.378	0.556	**0.613**
C Chen	0.508	0.226	0.538	0.266	0.407	**0.475**
D Johnson	0.626	0.555	0.639	0.620	**0.760**	0.596
J Martin	0.816	0.562	0.790	0.697	**0.844**	**0.738**
J Robinson	0.651	0.570	0.683	0.594	**0.735**	**0.625**
J Smith	0.617	0.642	0.733	0.758	0.686	0.745
K Tanaka	0.683	0.534	0.699	0.570	0.665	0.461
M Brown	0.725	0.629	0.786	0.759	**0.803**	**0.774**
M Jones	0.655	0.623	0.715	0.679	**0.830**	0.469
M Miller	0.820	0.811	0.881	0.887	**0.911**	0.857
Avg	0.673	0.583	0.712	0.63	**0.728**	**0.642**
Improvement %	Over HHC				**8.17**	**10.12**
	Over SAND				**2.25**	**1.91**

Table 22 and Figure 9 show that our method outperforms HHC by 8.17% in k-measure and 10.12% in f-measure; and SAND by 2.25% in k-measure and 1.91% in f-measure. Table 22 reports k-measure and f-measure values against each dataset of DBLP collection for each method, whereas Figure 9 shows a comparison of SHC with baselines for average values of all DBLP datasets.

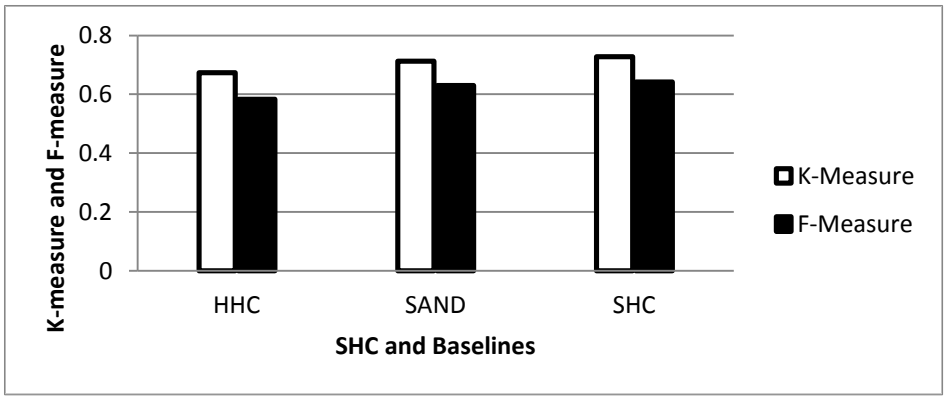

Figure 9: Comparison of SHC with baseline methods on DBLP datasets

SHC results on DBLP datasets are not as favorable as those of BDBComp results. There are two reasons. The *first* reason is that in these datasets the ambiguous author names are in short form (first name initial and last name) and our name similarity measure does not have such impact on short names as on full names (BDBCopm datasets). The *second* reason is the nature of SHC algorithm. In some cases, it wrongly merges cluster *ks* (the smallest cluster) of $\mathcal{B}$ with one of first few clusters of $\mathcal{B}$ say, $k_1, k_2 \ldots$ (the bigger clusters) without investigating all other clusters of $\mathcal{B}$ bigger than *ks*. There may be any other more probable cluster to be merged with cluster *ks*. If this investigation is performed, of course, this is possible, the algorithm will take much more time that may not be affordable.

5.5.3. Comparison of Similarity Measures

Table 23 illustrates k-measure and f-measure values for different similarity measures employed in SHC algorithm on BDBComp datasets. Table 23 shows that our proposed similarity measures collectively outperform baseline measures by 16.2 % in k-measure and 14.2% in f-measure. Equation **9** and CAM collectively outperform baseline measures by 12.1 % in k-measure and 9.1% in f-measure. These values depict that Equation **7** alone improves k-measure by 4.1% and f-measure by 5.1%.

Table 23: Comparison between different similarity measures on BDBComp datasets

Ambiguous Author Name	(Jaccord for Author Name and Co-author, Cosine for Title and Venue)		(FCM for Author Name, Eq. 9 for Co-authors and CAM for Title and Venue)		(Eq 7 Author Name, Eq. 9 for Co-authors and CAM for Title and Venue)	
	K-Measure	**F-Measure**	**K-Measure**	**F-Measure**	**K-Measure**	**F-Measure**
A. Oliveira	0.581	0.606	**0.90**	**0.856**	**0.922**	**0.911**
A. Silva	0.762	0.785	**0.873**	**0.88**	**0.969**	**0.962**
F. Silva	0.876	0.861	**0.98**	**0.978**	0.98	0.978
J. Oliveira	0.773	0.792	**0.878**	**0.863**	**0.919**	**0.937**
J. Silva	0.829	0.808	**0.871**	**0.861**	**0.893**	**0.878**
J. Souza	0.655	0.735	**0.842**	**0.779**	**0.951**	**0.923**
L. Silva	0.903	0.880	**0.969**	**0.947**	**0.970**	0.947
M. Silva	0.893	0.889	**0.926**	**0.914**	**0.976**	**0.97**
R. Santos	0.975	0.971	0.975	0.971	0.975	0.971
R. Silva	0.944	0.934	**0.963**	**0.957**	0.963	0.957
Average	0.819	0.826	**0.918**	**0.901**	**0.952**	**0.943**
Improvement %	-	-	**12.1**	**9.1**	**16.2**	**14.2**

Table 23 and Table 24 show that SHC algorithm outperforms baseline methods, even if we use the similarity measures employed by baselines. We investigate this fact on BDBComp datasets. Table 24 shows that SHC algorithm outperforms HHC by 12.6% and 44.02% in k-measure and f-measure respectively. When comparing with SAND, SHC algorithm's k-measure declines by 3.18%, but on the other hand its f-measure exceeds than SAND by 43.76%. Resultantly SHC algorithm outperforms both baseline methods.

Table 24: Percentage improvement of SHC algorithm over baseline methods on BDBComp datasets

Percentage Improvement	HHC		SAND	
	K-Measure	**F-Measure**	**K-Measure**	**F-Measure**
Commutative % improvement of proposed clustering algorithm and proposed Similarity Measures	28.8	58.22	13.02	57.96
% improvement of our proposed Similarity Measures	16.2	14.20	16.20	14.20
% improvement of proposed clustering algorithm	12.6	44.20	- 3.18	43.76

We investigate the impact of our proposed SDK index in AND process on DBLP-ref datasets. Shoaib index and SDK index are useful only if two text segments (or text documents) have the term frequency of some words greater than 1. This is very rare to happen in title and venue attributes. That is why we do not investigate it for title and venue attributes. BDBComp and DBLP collections do not have the attributes having long

text segments like ref-titles. That is why we cannot investigate its effect in these datasets. We choose cosine to compare with SDK index because it is the most used similarity measure in AND research works [106] [104]. Table 25 reports the results generated for SDK index and cosine measure for ref-titles attribute. SDK index outperforms cosine measure by 1.2% and 2.7% in k-measure and f-measure respectively. We also employ Shoaib index for ref-titles attribute in DBLP-Ref datasets. This measure provides the same results as those of SDK index.

Table 25: Comparison of SDK index and cosine measure on ref-titlles attribute

Ambiguous Authors	**Cosine (at threshold 0.1)**		**SDK Index (at threshold 0.15)**	
	K-measure	**F-measure**	**K-measure**	**F-measure**
A Gupta	0.985272	0.727273	0.985272	0.727273
B Liu	0.726354	0.685714	0.726354	0.685714
C Chang	1.0	1.0	1.0	1.0
H Fang	0.692715	0.666667	**0.746264**	**0.685714**
J Smith	0.865692	0.685714	0.865692	0.685714
K Zhagn	0.755929	0.8	0.755929	0.8
R Kumar	0.865812	0.606061	**0.885989**	**0.729167**
Average	0.842	0.739	**0.852**	**0.759**
Improvement %			**1.2**	**2.7**

For cosine, 0.1 is the best threshold and, for SDK index it is 0.2. By using SDK index we achieve 1.2% improvement in k-measure and 2.7% in f-measure. We also tested the Shoaib index and it gave the same results for each dataset at the same threshold value (0.2) as that for SDK index. SDK index varies only when there is a higher difference in term frequency between two documents otherwise both (Shoaib index and SDK index) show similar behavior and impact.

5.5.4. Impact of Ref-titles in AND Process

To prove our notion that references may improve AND performance. We exploit SHC algorithm and our proposed similarity measures on DBLP-Ref datasets. Here we use only titles part of references (ref-titles). We do not use ref-coauthors attribute because we have investigated (in chapter 4) that ref-coauthors have comparatively low pair wise similarity as compared to that of ref-titles'. Ref-venues may have some impact in AND process but

we investigate only ref-titles. Table 26 reports values of k-measure and f-measure with and without ref-titles on DBLP-Ref datasets. Using ref-titles attribute improves k-measure by 0.6% and f-measure by 8%.

Table 26: Performance with and without Ref-Titles on DBLP-Ref datasets

Ambiguous Author Name	Without Ref-titles		With Ref-titles	
	K-Measure	F-Measure	K-Measure	F-Measure
A Gupta	0.957	0.686	**0.985**	**0.727**
B Liu	0.717	0.571	**0.726**	**0.686**
C Chang	1.0	1.0	1.0	1.0
H Fang	0.746	0.686	0.746	0.686
J Smith	0.866	0.686	0.866	0.686
K Zhang	0.756	0.8	0.756	0.8
R Kumar	0.888	0.491	0.888	**0.729**
Avg.	0.847	0.703	**0.852**	**0.759**
Improvement %			**0.6**	**8**

We experimented AND process for different combinations of threshold values for different attributes, each ranging from 0.05 to 0.95 with an increment of 0.05. The best combination of threshold values for our approach is 0.75 for ambiguous author name, 0.4 for title, 0.6 for venue and 0.2 for ref-titles attributes.

5.6. Chapter Summary

We have proposed clustering algorithm for author name disambiguation. Our proposed clustering algorithm resolves name disambiguation problem in two phases. Phase I exploits only co-authors attribute, the most informative in AND process, and phase II uses all other available sources of information. Phase II also helps decrease fragmentation. We employ our own proposed similarity measures for all attributes. Our measures, on DBDComp datasets on the average, outperform base lime similarity measures by 16.2% in k-measure and 14.20A% in f-measure

Experiments show that our proposed clustering algorithm's performance (k-measure and f-measure) is better than baseline methods. For example, on DBDComp collection it is 12.6% and 44.20% better than HHC in k-measure and f-measure respectively. For the same data collection, it also outperforms SAND by 43.76% in f-measure, but it shows a loss of 3.18% in k-measure. We have also employed ref-titles attribute as additional source of information in our clustering algorithm. This attribute improves f-measure by 0.6% and

k-measure by 8% on DBLP-Ref collection. Our proposed clustering algorithm is capable to assign new citation to the actual cluster on the base of previous citations.

Chapter 6. Summary and Future Directions

Chapter 6

Summary and Future Directions

In this chapter we summarize our research work findings and also point out some future directions.

6.1. Summary

We propose different similarity measures for comparing publications in a pair wise fashion. Each proposed similarity measure is for different types of data. Equation 7 estimates similarity between two names; equation 9 between co-authors of two citations; equation 12 between two titles or venues; and Shoaib index (equation 13) and SDK index between two ref-titles of two publications. Equation 12 is absolute measure but it is usable only for data where term frequency is not greater than 1. Shoaib index and SDK index are very close to absolute measures. Equation 7 and 9 satisfy our assumptions I, II and III and improve AND performance. Our proposed measures can be applied to any type of textual data where name entities (not necessarily human names) or textual data or both exist.

SDK index shows behavior closer to our assumptions in all scenarios discussed in the results section of chapter 3. Its output is nearer to absolute similarity value between two documents. It is, in some cases, equal to Shoaib index, Dice and Information Theoretic; and, in some cases it is better (closer to absolute) than these measures. We have compared the behavior of six well known similarity measures. Out of these, cosine measure is the farthest from absolute similarity value and SDK index is the nearest. Cosine shows inverse behaviors in certain conditions while Shoaib index and SDK index don't. Our proposed measures need not any information about the number of documents in the collection as it is needed in TFIDF based similarity functions. Shoaib and SDK

indices are basically designed for textual documents. They can also be applied for entity names.

We also propose to utilize references as sources of information for publications. We have empirically proved that references attribute (ref-titles and ref-coauthors) help improve publications similarity and also improve AND performance. Experiments show that references attributes provide a good source of similarity information for publications. Ref-titles attribute improves k-measure by 0.6% and f-measure by 8% on DBLP-Ref datasets. Ref-coauthors attribute, though not more informative than co-authors attribute yet provides a reasonable amount of similarity information. From this discussion, it is concluded that the proposed idea of exploiting references for estimating academic document similarity is worthwhile.

Experiments show that our methodology outperforms baseline methods. Our methodology (algorithm and similarity measures) outperforms, on the average, HHC by 28.8% in k-measure and 58.22 % in f-measure, and SAND by 13.02% in k-measure and 57.96% in f-measure on BDBComp datasets. It outperforms, on the average, HHC by 8.17% in k-measure and 10.12 % in f-measure; and SAND by 2.25% in k-measure and 1.91% in f-measure on DBLP datasets.

6.2. Future Directions

In future we will compare our similarity measures with other similarity measures on different AND methods. Our similarity measures can be employed and compared with other similarity measures in the fields where document similarity is the main focus. We will apply our AND clustering algorithm along with references information for academic document clustering. We are also interested to employ ref-coauthors and ref-venue attributes to analyze their impact on publication similarity. Ref-coauthors and ref-venue can be empirically tested whether they have any effect on similarity of publications.

Our proposed similarity measures for name similarity and co-authors similarity (equation 7 and 9) are better options as they depict our assumptions I to III and improve AND performance, but they are not able to handle frequency of a name greater than 1. A similarity measure that can handle higher frequency and assigning proper weight to

frequent terms can be devised. We are also interested to find the net effect of choosing seed by our method over random selection of seed.

Our clustering algorithm works on the notion if two publications are more similar then it is more probable that they are from the same person, but what if an author changes his/her field of research. A methodology can be devised to handle this issue. It is, perhaps, possible by dividing the cluster (citations) of an author to sub-clusters on the base of time of publication and the similarity values of different attributes. More similar publications written within a specific period can be grouped in a sub-cluster. In this way multiple sub-clusters will get the same author label. New citation can be compared to each sub-cluster and be assigned to the most appropriate one.

We are also interested to exploit semi-supervised learning approach for AND problems. Our algorithm 1 is capable of increasing or decreasing precision of constructed clusters dynamically by increasing or decreasing the value of threshold. We can construct very pure clusters by fixing the high threshold value. The citations showing low similarity value with other citations may be left undecided, i.e., they are not assigned to any cluster. The pure clusters can be used as training examples for undecided citations. This high threshold value on the other hand may cause fragmentation. Our clustering algorithm is also capable to resolve the problem of fragmentation in an efficient way.

References

References

[1] J. Kleb and R. Volz, "Ontology-based entity disambiguation with natural language patterns," in *4th International Conference on Digital Information Management*, pp. 18-25, 2009.

[2] N. Aswani, K. Bontcheva and H. Cunningham, "Mining information for instance unification," in *5th International Semantic Web Conference, Berlin Heidelberg*, pp. 329-342, 2006.

[3] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *14th International Conference on World Wide Web*, pp. 463–470, 2005.

[4] J. Tang, A. Fong, B. Wang and J. Zhang, "A unified probabilistic framework for name disambiguation in digital library," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 975-987, 2012.

[5] A. P. d. Carvalho, A. A. Ferreira, A. H. Laender and M. A. Gonçalves, "Incremental unsupervised name disambiguation in cleaned digital libraries," *Journal of Information and Data Management*, vol. 2, no. 3, pp. 289-304, 2011.

[6] A. A. Ferreira, M. A. Gonçalves and A. H. F. Laender, "A brief survey of automatic methods for author name disambiguation," *ACM SIGMOD Record*, vol. 41, no. 2, pp. 15-26, 2012.

[7] M. Ley, "The DBLP computer science ibliography, evolution, research issues, perspectives," in *9th Iternational Symposyum on String Processing and Information Retrieval*, pp. 1-10, 2002.

[8] C. L. Giles, K. D. Bollacker and S. Lawrence, "Citeseer: an automatic citation indexing system," in *3rd ACM Conference on Digital Libraries*, pp. 89-98, 1998.

[9] H. Han, H. Zha and L. Giles, "Name disambiguation in author citations using a k-way spectral clustering method," in *ACM/IEEE Joint Conference on Digital Libraries*, pp. 334-343, 2005.

[10] B. Malin, "Unsupervised name disambiguation via social network similarity," in *Workshop on Link Analysis, Counterterrorism and Security in conjunction with the SIAM International Conference on Data Mining*, pp. 93-102, 2005.

[11] X. Yin, J. Han and P. S. Yu, "Object distinction: distinguishing objects with identical names," in *International Conference on Data Engineering*, pp. 311-318, 2007.

[12] D. Lee, B. On, J. Kang and S. Park, "Effective and scalable solutions for mixed and split citation problems in digital libraries," in *2nd Workshop on Information Quality in Informational Systems*, pp. 69-76, 2005.

[13] Y. F. Tan, M.-Y. Kan and D. Lee, "Search engine driven author disambiguation," in *6th ACM/IEEE Joint Conference on Digital libraries*, pp. 314-315, 2006.

[14] L. Shu, B. Long and W. Meng, "A latent topic Model for complete entity resolution," in *IEEE 25th International Conference on Data Engineering*, pp. 880-891, 2009.

[15] I. Bhattacharya and L. Getoor, "A latent dirichlet model for unsupervised entity resolution," in *SIAM Conference on Data Mining*, pp. 33-42, 2006.

[16] C. Galvez and F. Moya-Aneg´on, "Approximate personal name-matching through finite-state graphs," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 13, pp. 1960–1976, 2007.

[17] P. Treeratpituk and C. Giles, "Disambiguating authors in academic publications using random forests," in *9th ACM/IEEE Joint Conference on Digital Libraries*, pp. 39-48, 2009.

[18] B. W. On, I. Lee and D. Lee, "Scalable clustering methods for the name disambiguation problem," *Knowledge and Information Systems*, vol. 31 , no. 1, pp. 129-151, 2012.

[19] X. Fan, J. Wang, X.Pu, L. Zhou and B. Lv, "On graph-based name disambiguation," *ACM Journal of Data and Information Quality*, vol. 2, no. 2, pp. 1-23, 2011.

[20] L. Branting, "A comparative evaluation of name-matching algorithms," in *International Conference on Artificial Intelligence and Law*, pp. 224-232, 2003.

[21] V. I. Torvik and N. R. Smalheiser, "Author name disambiguation in medline," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 3, 2009.

[22] A. A. Ferreira, A.Velosol, M. A. Gonçalves and A. H. F. Laender, "Effective self-training author name disambiguation in scholarly digital libraries," in *10th ACM/IEEE Joint Conference on Digital Libraries*, pp. 39-48, 2010.

[23] M. Shoaib, A. Daud and M. S. H. Khiyal, "Role of references in similarity estimation of publications," *The International Arab Journal of Information Technology*, vol. 13, no. 3, In Press.

[24] A. A. Ferreira, A. Veloso, M. A. Gonçalves and A. H. Laender, "Self-training

author name disambiguation for information scarce scenarios," *Journal of the Association for Information Science and Technology,* vol. 65, no. 6, pp. 1257-1278, 2014.

[25] R. G. Cota, M. A. Gon˜A˘galves and A. H. F. Laender, "A heuristic-based hierarchical clustering method for author name disambiguation," in *Brazilian Symposium on Data Base*, pp. 20–34, 2007.

[26] N. R. Smalheiser and V. I. Torvik, "Author name disambiguation," *Annual Review of Information Science and Technolog,* vol. 43, no. 1, pp. 1–43, 2009.

[27] C. L. Scoville, E. D. Johnson and A. L. McConnell, "When a rose is not a rose: the vagaries of author searching," *Medical Reference Services Quarterly,* vol. 22, no. 4, pp. 1–11, 2003.

[28] H. Köpcke, A. Thor and E. Rahm, "Evaluation of entity resolution approaches on real world match problems," *the VLDB Endowment,* vol. 3, no. 1, pp. 484-493, 2010.

[29] H. Han, L. Giles, H. Zha, C. Li and K. Tsioutsiouliklis, "Two supervised learning approaches for name disambiguation in author citations," in *ACM/IEEE joint Conference on Digital Libraries*, pp. 296–305, 2004.

[30] V. I. Torvik, M. Weeber, D. R. Swanson and N. R. Smalheiser, "A probabilistic similarity metric for medline records a model for author name disambiguation," *Journal of the American Society for Information Science and Technology,* vol. 56, no. 2, pp. 140-158, 2005.

[31] A. Culotta, P. Kanani, R. Hall, M. Wick and A. McCallum, "Author disambiguation using error-driven machine learning with a ranking loss function," in *Sixth International Workshop on Information Integration on the Web*, pp. 289-295, 2007.

[32] Y. Qian, Y. Hu, J. Cui, Q. Zheng and Z. Nie, "Combining machine learning and human judgment in author disambiguation," in *20th ACM International Conference on Information and Knowledge Management*, pp. 1241-1246, 2011.

[33] E. de Souza, A. Ferreira and M. Gonçalves, "Combining classifiers and user feedback for disambiguating author names," in *15th ACM/IEEE-CE Joint Conference on Digital Libraries*, New York, USA, 2015.

[34] M. Song, E. H. Kim and K. H. J., "Exploring author name disambiguation on PubMed-scale," *Journal of Informetrics*, vol. 9, no. 4, pp. 924-41, 2015.

[35] A. F. Santana, M. A. Gonçalves, L. A. H. and A. A. Ferreira, "On the combination

of domain-specific heuristics for author name disambiguation: the nearest cluster method," *International Journal on Digital Libraries,* vol. 16, no. 3-4, pp. 229-246, 2015.

[36] J. W. A. Oliveira, A. H. F. Laender and M. A. Gonçalves, "Remoção de ambigüidades na Identificação de Autoria de Objetos Bibliográficos," in *Simpósio Brasileiro de Banco de Dados*, pp. 205-219, 2005.

[37] G. Salton, A. Wong and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM,* vol. 18 , no. 11, pp. 613-620, 1975.

[38] V. Vapnik, The nature of statistical learning theory, 2nd ed., Springer-Verlag, 1995.

[39] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines," USA: Cambridge University Press, 2000.

[40] H. Han, H. Zha and C. L. Giles, "A model-based k-means algorithm for name disambiguation," in *2nd International Semantic Web Conference, Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data,* pp. 710-715, 2003.

[41] F. Wang, J. Li, J. Tang, J. Zhang and K. Wang, "Name disambiguation using atomic clusters," in *Web-Age Information Management*, pp. 357-364, 2008.

[42] F. Wang, J. Tang, J. Li and K. Wang, "A constraint-based topic modeling approach for name disambiguation," *Frontiers of Computer Science, China,* vol. 4, no. 1, pp. 100-111, 2010.

[43] D. Zhang, J. Tang, J. Li and K. Wang, "A constraint-based probabilistic framework for name disambiguation," in *ACM Conference on Information and Knowledge Management*, pp. 1019–1022, 2007.

[44] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research,* vol. 3, pp. 993–1022, 2003.

[45] A. E. Gelfand, "Gibbs sampling," *Journal of the American Statistical Association,* vol. 95, no. 452, pp. 1300–1304, 2000.

[46] X. Sun, J. Kaur, L. Possamai and F. Menczer, "Detecting ambiguous author names in crowdsourced scholarly data," in *3rd IEEE Conference on Social Computing*, pp. 569-571, 2011.

[47] D. T. Hoang, Kaur and F. Menczer, "Crowdsourcing scholarly data," in *Web Science Conference: Extending the Frontiers of Society*, pp. 465-474, 2010.

[48] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[49] H. Han, W. Xu, H. Zha and C. L. Giles, "A hierarchical naive Bayes mixture model for name disambiguation in author citations," in *ACM Symposium on Applied Computing*, pp. 1065–1069, 2005.

[50] B. W. On, E. Elmacioglu, D. Lee, J. Kang and J. Pei, "An effective approach to entity resolution problem using quasi-clique and its application to digital libraries," in *ACM /IEEE Joint Conference on Digital Libraries*, pp. 51–52, 2006.

[51] Y. Song, J. Huang and I. G. Councill, "Efficient topic-based unsupervised name disambiguation," in *ACM/IEEE Joint Conference on Digital libraries*, pp. 18–23, 2007.

[52] I. Bhattacharya and L. Getoor, "Collective entity resolution in relational data," *ACM Transactions on Knowledge Discovery from Data,* vol. 1, no. 1, article 5, 2007.

[53] B. W. On and D. Lee, "Scalable name disambiguation using multi-level graph partition," in *SIAM Conference on Data Mining*, pp. 178-189, 2007.

[54] J. M. Soler, "Separating the articles of authors with the same name," *Scientometrics,* vol. 72, no. 2, pp. 281–290, 2007.

[55] K. H. Yang, H.-T. Peng, J.-Y. Jiang, H.-M. Lee and J.-M. Ho, "Author name disambiguation for citations using topic and web correlation," in *European Conference on Research and Advanced Technology for Digital Libraries*, pp. 185–196, 2008.

[56] I. S. Kang, S.-H. Na, S. Lee, H. Jung, P. Kim, W.-K. Sung and J.-H. Lee, "On co-authorship for author disambiguation," *Information Processing & Management,* vol. 45, no. 1, pp. 84–97, 2009.

[57] D. A. Pereira, B. A. Ribeiro-Neto, N. Ziviani, A. H. F. Laender, M. A. Goncalves and A. A. Ferreira, "Using web information for author name disambiguation," in *ACM/IEEE Joint Conference on Digital Libraries*, pp. 49–58, 2009.

[58] Y. Qian, Q. Zheng, T. Sakai, J. Ye and J. Liu, "Dynamic author name disambiguation for growing digital libraries," *Information Retrieval Journal,* vol. 18, no. 5, pp. 379-412., 2015.

[59] Y. Liu and Y. Tang, "Network based framework for author name disambiguation applications," *International Journal of u-and e-Service, Science and Technology,*

vol. 8, no. 9, pp. 75-82, 2015.

[60] T. Arif, "Exploring the use Of hybrid similarity measure for author name disambiguation," *IONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH* , vol. 4, no. 12, pp. 171-175, 2015.

[61] T. Arif, R. Ali and M. Asger, "A multistage hierarchical method for author name disambiguation," *International Journal of Information Processing*, vol. 9, no. 3, pp. 92-105, 2015.

[62] W. R. Gilks, S. Richardson and D. J. Spiegelhalter, Markov Chain Monte Carlo in Practice, London: Chapman and Hall, 1996.

[63] A. Dempster, N. Laird and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B.,* vol. 39, no. 1, pp. 1–38, 1977.

[64] T. Hofmann, "Probabilistic latent semantic analysis," in *15th Annual Conference on Uncertainty in Artificial Intelligence*, Stockholm, pp. 418-429, 1999.

[65] D. Shin, T. Kim, H. Jung and J. Choi, "Automatic method for author name disambiguation using social networks," in *24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 1263-1270, 2010.

[66] D. Shin, J. Kang, J. Choi and J. Yang, "Detecting collaborative fields using social networks," in *Fourth International Conference on Networked Computing and Advanced Information Management*, pp. 325-328, 2008.

[67] K. Yang and Y. Wu, "Author name disambiguation in citations," in *Web Intelligence and Intelligent Agent Technology*, pp. 335 – 338, 2011.

[68] M. Levin, S. Krawczyk, S. Bethard and D. Jurafsky, "Citation-based bootstrapping for large-scale author disambiguation," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 5, pp. 1030-1047, 2012.

[69] A. A. Ferreira, R. Silva, M. A. Gonçalves, A. Veloso and A. H. Laender, "Active associative sampling for author name disambiguation," in *12th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 175-184, 2012.

[70] B. On, D. Lee, J. Kang and P. Mitra, "Comparative study of name disambiguation problem using scalable blocking-based framework," in *ACM/IEEE Joint Conference on Digital Libraries*, pp. 344-353, 2005.

[71] J. Huang, S. Ertekin and C. L. Giles, "Efficient name disambiguation for large scale databases," in *European Conference on Principals of Data Mining and Knowledge*

Discovery, pp. 536-544, 2006.

[72] B. W. On, E. Elmacioglu, D. Lee, J. Kang and J. Pei, "Improving grouped-entity resolution using quasi-cliques," in *IEEE International Conference on Data Mining*, pp. 12-21, 2006.

[73] G. Louppe, H. Al-Natsheh, M. Susik and E. Maguire, "Ethnicity sensitive author disambiguation using semi-supervised learning," *arXiv preprint arXiv*, 2015.

[74] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[75] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar and S. Fienberg, "Adaptive name-matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16-23, 2003.

[76] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson and P. Singla, "Markov logic," in *Probabilistic Inductive Logic Programming*, vol. 4911, 2008, pp. 92-117.

[77] A. Culotta and A. McCallum, "Practical Markov logic containing first-order quantifiers with application to identity uncertainty," in *Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pp. 41-48, 2006.

[78] L. Getoor and B. Taskar, Introduction to Statistical Relational Learning, Cambridge: MIT Press, 2007.

[79] M. Song and A. Rudniy, "Detecting duplicate biological entities using Markov random field-based edit distance," *Knowledge and Information Systems*, vol. 25, no. 2, pp. 371-387, 2010.

[80] X. Yu and W. Lam, "Probabilistic joint models incorporating logic and learning via structured variation approximation for information extraction," *Knowledge and Information Systems*, vol. 32, no. 2, pp. 415-444, 2012.

[81] P. Singla and P. Domingos, "Entity resolution with Markov logic," in *Sixth International Conference on Data Mining*, pp. 572-582, 2006.

[82] Z. Ghahramani and M. Jordan, "Factorial Hidden Markov Models," *Machine Learning*, vol. 29, no. 2-3, pp. 245-273, 1997.

[83] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771-1800, 2002.

[84] L. Huang, D. Milne, E. Frank and I. H. Witten, "Learning a concept-based document similarity measure," *Journal of the American Society for Information*

Science and Technology, vol. 63 , no. 8, pp. 1593-1608, 2012.

[85] A. A. Ferreira, T. M. Machado and M. A. Gonçalves, "Improving author name disambiguation with user relevance feedback," *Journal of Information and Data Management,* vol. 3, no. 3, pp. 332-347, 2012.

[86] A. Strotmann, D. Zhao and T. Bubla, "Author name disambiguation for collaboration network analysis and visualization," *American Society for Information Science and Technology*, vol. 46, no. 1, pp. 1-20, 2009.

[87] D. V. Kalasnikov and S. Mehrotra, "Domain-independent data cleaning via analysis of entity relationship graph," *ACM Transactions on Data Systems,* vol. 31, no. 2, pp. 716–767, 2006.

[88] Z. Chen, D. V. Kalashnikov and S. Mehrotra, "Adaptive graphical approach to entity resolution," in *ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 204–213, 2007.

[89] H. Jin, L. Huang and P. Yuan, "Name disambiguation using semantic association clustering," in *IEEE International Conference on e-Business Engineering*, pp. 42-48, 2009.

[90] J. Pei, D. Jiang and A. Zhang, "On mining cross-graph quasi-cliques," in *International Conference on Knowledge Discovery and Data Mining*, pp. 228-238, 2005.

[91] D. M. McRae-Spencer and N. R. Shadbolt, "Also by the same author AKTiveAuthor, a citation graph approach to name disambiguation," in *6th ACM/IEEE-CS Joint Conference on Digital libraries*, pp. 53-54, 2006.

[92] X. Wang, J. Tang, H. Cheng and P. Yu, "ADANA: Active name disambiguation," in *IEEE International Conference on Data Mining*, pp. 794 – 803, 2012.

[93] M. Yankova, H. Saggion and Cunningham, "Adopting ontologies for multisource identity resolution," in *First International Workshop on Ontology-supported Business*, article 6, 2008.

[94] H. T. Nguyen and T. Cao, "Named entity disambiguation on ontology enriched by Wikipedia," in *IEEE International Conference on Research, Innovation and Vision for the Future*, pp. 247-254, 2008.

[95] H. Nguyen and T. Cao, "Enriching ontologies for named entity disambiguation," in *4th International Conference on Advances in Semantic Processing*, pp. 37-42, 2010.

[96] D. Song and J. Heflin, "Domain-independent entity coreference for linking ontology instances," *Journal of Data and Information Quality (JDIQ),* vol. 4, no. 2, pp. article 7, 2013.

[97] J. Hassell, B. Aleman-Meza and I. B.Arpinar, "Ontology-driven automatic entity disambiguation in unstructured text," in *5th International Semantic Web Conference*, pp. 44-57, 2006.

[98] Y. Park and J. Kim, "OnCU system: ontology-based category utility approach for author Nnme disambiguation," in *2nd International Conference on Ubiquitous Information Management and Communication*, pp. 63-68, 2008.

[99] M. Rafi and M. S. Shaikh, "An improved semantic similarity measure for document clustering based on topic maps," *Sindh University Research Journal (Science Series),* vol. 45, no. A-1, pp. 59-64, 2013.

[100] A. Strehl, J. Ghosh and R. Mooney, "Impact of similarity measures on web-page clustering," in *AAAI-2000 Workshop on Artificial Intelligence for Web Search*, pp. 58-64, 2000.

[101] A. Daud, L. Z. J. Li and F. Muhammad, "Knowledge discovery through directed probabilistic topic models, a survey," *Frontiers of Computer Science in China,* vol. 4, no. 2, pp. 280-301 , 2010.

[102] M. Donald, D. Susan and M. Christopher, "Similarity measures for short segments of text," in *European Confrence on information Retrieval*, pp. 16-27, 2007.

[103] X. Wan, "A novel document similarity measure based on earth mover's distance," *Information Sciences,* vol. 177, no. 18, pp. 3718–3730, 2007.

[104] M. Shoaib, A. Daud and M. S. H. Khiyal, "Improving similarity measures for publications with soecial focus on author name disambiguation," *Arabian Journal for Science and Engineering,* vol. 40, no. 6, 1591-1605, 2015.

[105] M. Shoaib, A. Daud and M. S. H. Khiyal, "An improved similarity measure for text documents," *Journal of Basic and Applied Scientific Resarch,* vol. 4, no. 6, pp. 215-223, 2014.

[106] T. Kazem and V. Rushikesh, "Effects of similarity metrics on document clustering," in *Seventh International Conference on Information Technology*, pp. 222-226, 2010.

[107] A. Huang, "Similarity measures for text document clustering," in *New Zealand Computer Science Research Student Conference*, pp. 49-56, 2008.

[108] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 16-22, 1999.

[109] S. Pandit and S. Gupta, "A comparative study on distance measuring approaches for clustering," *International Journal of Research in Computer Science,* vol. 2, no. 1, pp. 29-31, 2011.

[110] W. Cohen, P. Ravikumar and S. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *IJCAI-03, the Workshop on Information Integration on the Web*, pp. 73-78, 2003.

[111] M. D. Lee, B. Pincombe and M. Welsh, "An empirical evaluation of models of text document similarity," in *XXVII Annual Conference of the Cognitive Science Society*, pp. 1254-1259, 2005.

[112] M. Shoaib, M. N. Yasin, H. Niazi, M. I. Saeed and S. H. Khiyal, "Relational WordNet model for semantic search in Holy Quran," in *International Conference on Emerging Technologies (IEEE ICET 09)*, pp. 29-35, 2009.

[113] R. Mercer and C. D. Marco, "A design methodology for a biomedical literature indexing tool using the rhetoric of science," in *BioLink workshop in conjunction with Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, pp. 77–84, 2004.

[114] H. Nanba, N. Kando and M. Okumura, "Towards multi paper summarization using reference information," in *16th International Joint Conferences on Artificial Intelligence*, pp. 926–931, 1999.

[115] B. Aljaber, N. Stokes, J. Bailey and J. Pei, "Document clustering of scientific texts using citation contexts," *Information Retrieval,* vol. 13, no. 2, pp. 101-131, 2010.

[116] H. Jeon, "A reference comments crawler for assisting research paper writing," *The International Arab Journal of Information Technology*, pp. 27-33, 2013.

[117] R. G. Cota, A. A. Ferreira, M. A. Gonçalves, A. H. F. Laender and C. Nascimento, "An unsupervised heuristic-based hierarchical method for name Disambiguation in bibliographic Citations," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 9, pp. 1853–1870, 2010.

[118] P. Kanani, A. McCallum and C. Pal, "Improving author coreference by resource-bounded information gathering from the web," in *International Joint Conferences on Artificial Intelligence*, pp. 429-434, 2007.

[119] F. H. Levin and C. A. Heuser, "Evaluating the use of social networks in author

name disambiguation in digital libraries," *Journal of Information and Data Management,* vol. 1, no. 2, pp. 183–197, 2010.

[120] A. Veloso, A. A. Ferreira, M. A. Gonçalves, A. H. F. Laender and W. J. Meira, "Cost-effective on-demandaAssociative author name disambiguation," *Information Processing &. Management,* vol. 48, no. 4, pp. 680– 697, 2012.

[121] J. A. Aslam and M. Frost, "An information-theoretic measure for document similarity," in *26th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp. 449–450, 2003.

[122] H. U. Khan, S. M. Saqlain, M. Shoaib and M. Sher, "Ontology-based semantic search in Holy Quran," *International Journal of Future Computer and Communication,* vol. 2, no. 6, pp. 570-575, 2013.

[123] H. Dunn, "Record linkage," *American Journal of Public Health,* vol. 36, no. 12, pp. 1412-1416, 1946.

[124] D. Bitton and D. DeWitt, "Duplicate record elimination in large data files," *ACM Transactions on Database Systems,* vol. 8, no. 2, pp. 255-265, 1983.

[125] M. Hern´andez and S. Stolfo, "The merge/purge Problem for large database proceedings," in *International Conference on Management of Data ACM SIGMOD* , pp. 127-138, 1995.

[126] K. Cios, W. Pedrycz, R. Swiniarski and L. Kurgan, Data mining: A knowledge discovery approach, 1st ed., New York Inc., Cecaucus NJ: Springer-Verlag, 2003.

[127] W. Cohen, H. Kautz and D. McAllester, "Hardening soft information sources," in *6th International Conference on Knowledge Discovery and Data Mining*, pp. 255–259, 2000.

[128] A. Bagga, "Coreference, cross-document coreference and information extraction methodologies, Doctoral Dissertation," Duke University Durham, NC, USA, 1998.

[129] D. A. Dervos, N. Samaras, G. Evangelidis, J. Hyvärinen and Y. Asmanidis, "The universal author identifier system (UAI_Sys)," in *1st International Scientific Conference eRA: The Contribution of Information Technology in Science, Economy,Society and Education*, pp. 153-164, 2006.

[130] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *First Instructional Conference on Machine Learning*, pp. 353-356, 2003.

[131] M. M. Ghasemi and A. Mahjur, "A Method for Finding Similar Documents on the Basis of Repetition-Based Filtering," *Journal of Basic and Applied Scientific*

Research , vol. 3, no. 1, pp. 603-607, 2013.

[132] H. Azgomi and A. Mahjur, "A Solution for Calculating the False Positive and False Negative in LSH Method to Find Similar Documents," *Journal of Basic and Applied Scientific Research,* vol. 3, no. 1, pp. 466-472, 2013.

[133] H. Pasula, B. Marthi, B. Milch, S. J. Russell and I. Shpitser, "Identity uncertainty and citation matching," in *Neural Information Processing Systems: Natural and Synthetic*, pp. 1401-1408, 2002.

[134] G. Varelas, E. Voutsakis, P. Raftopoulou, E. G. Petrakis and E. E. Milios, "Semantic similarity methods in WordNet and their application to information retrieval on the Web," in *7th ACM International Workshop on Web Information and Data Management*, pp. 776-781, 2005.

Appendices

Appendix A: Screen Shorts

Application Front End

Figure 10 shows the front end of our AND application. HHC and SAND buttons are used for executing baseline methods and SHC for our proposed methodology. Lowest and highest threshold text boxes are used to adjust the minimum and maximum threshold values against each attribute. The list boxes (last column of AND application form) are used for selecting the similarity measures for respective attributes. All other items are self explanatory.

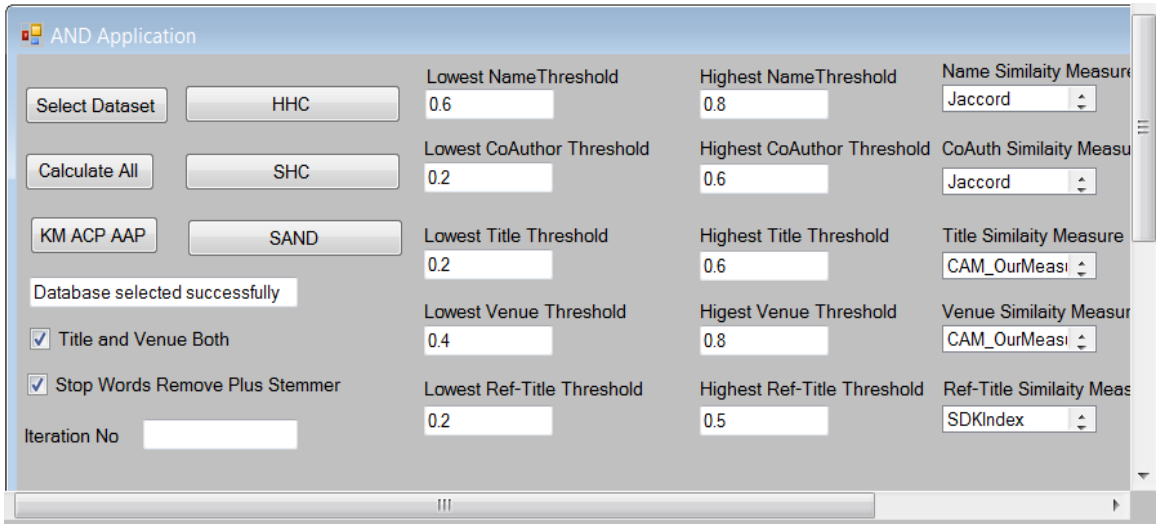

Figure 10: Application front end

Results File

Figure 11 shows the image of a text file containing the results produced on BDBComp collection for R Silva ambiguous author name. The text file records the summary of results including AND method used, threshold values, time consumed, ACP, AAP, k-measure and many other important things. This file also contains the detailed information of each and every empirical and theoretical cluster.

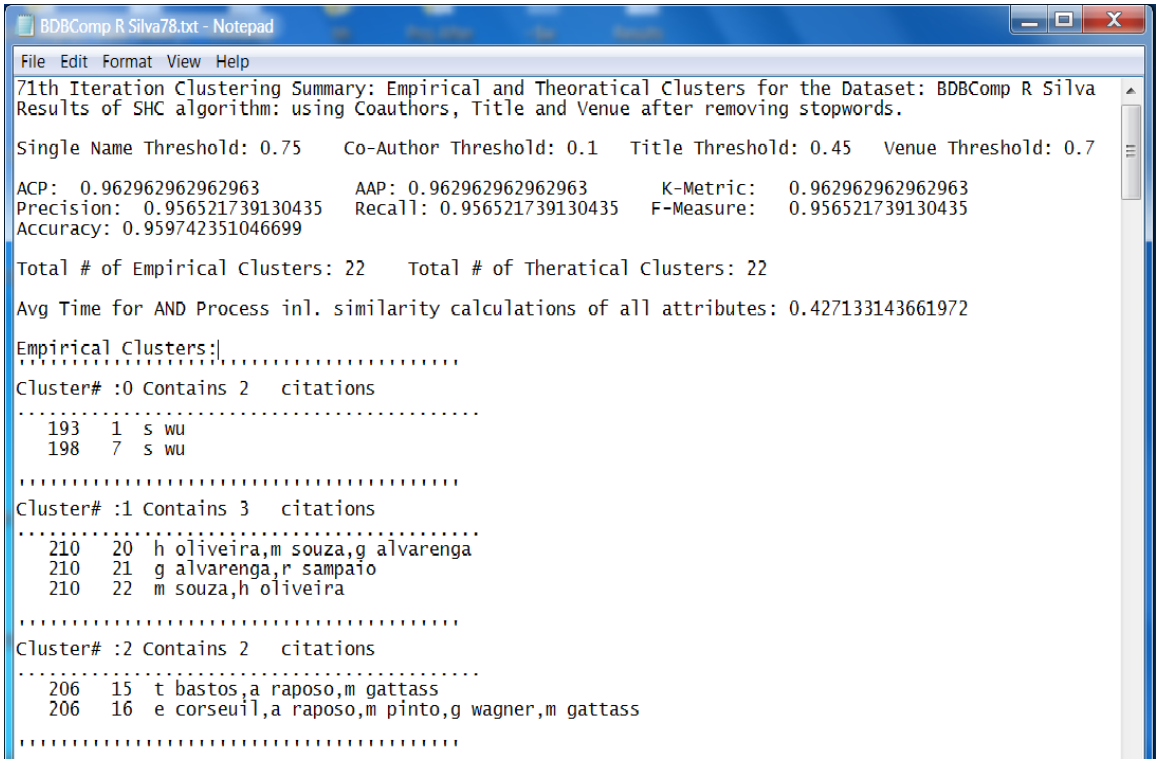

Figure 11: Results produced by SHC on BDBComp collection for R. Silva

Appendix B: Subsets of Dataset Tables

Subset of R Silva Dataset

Table 27 is selected as evidence for our assumption I and II. It consists of 27 records against 22 unique authors. The highlighted records are discussed in chapter 3 under the headings Assumption I and II. The records 9, 25 and 26 are completely highlighted whereas others (18-20) are partially highlighted. The former case is for assumption I and the latter is for assumption II.

Table 27: R Silva dataset, a subset of the BDBCommp collection

Author_ID	Publication #	Ambiguous Author Name
1	0	roseli maria da silva
2	1	r m da silva
3	2	rafael guilherme r da silva
4	3	ricardo pereira e silva
4	4	ricardo pereira e silva
5	5	roberta scaramussa da silva
6	6	romildo jose da silva
7	7	rosana marques da silva
8	8	rodrigo silva
9	9	rodrigo da silva
10	10	r i da silva
11	11	robson teixeira da silva
12	12	ricardo m a silva
13	13	rafael araujo silva
14	14	rogerio e da silva
15	15	romano j m da silva
16	16	romano j m silva
17	17	ricardo antonio camara da silva
18	18	ricardo m a silva
19	19	ricardo m a silva
20	21	ricardo m de a silva
20	22	ricardo m de a silva
20	20	ricardo m de a silva
21	23	renato a c silva
22	24	ronaldo p silva
23	25	roberto da silva
23	26	r da silva

Table 28 is selected as evidence for our assumption III. It consists of 20 records against 17 unique authors. The highlighted records are discussed in chapter 3 under the heading Assumption III.

Table 28: R Santos dataset, a subset of the BDBComp collection

Author_ID	Publication #	Co-Authors
174	0	j junior,m camargo,s gregorio,m ribeiro,f arruba-jr,o fernandes,w ishibashi
175	1	j leite,c klemtz,m mandel,a mantovani,s cintra
176	2	t ohashi,t yoshida,t ejima
177	3	a calsavara
178	4	i doi,j diniz,j swart,s santos
178	5	i doi,r teixeira,j diniz,j swart,s santos
179	6	f artola,s fontoura,m vellasco
180	7	j tambor,l paulino,a bazzan
181	8	a calazans,k oliveira
182	9	p adeodato,a arnaud,r cunha,g vasconcelos,d monteiro
183	10	f mesquita,a silva,e vilarinho
184	11	w caminhas,l errico
185	12	
186	13	r andrade,e marcal,c vidal,r rios
187	14	n medina,r lapa,m nero,a netto
188	15	p roberto,m goncalves,a laender
188	16	a silva,h santos,a laender,m goncalves
189	17	a peterle,c castro,c meffe,n bretas
190	18	j santos,j orozco
190	19	c hara