CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Determination of Feature Contribution Score and Feature Ranking in Author Name Disambiguation

by

Muhammad Aadil-ur-Rehman

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Computing
Department of Computer Science

2018

I would like to dedicate this work to my Father.

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD

CERTIFICATE OF APPROVAL

Determination of Feature Contribution Score and Feature Ranking in Author Name Disambiguation

by

Muhammad Aadil-ur-Rehman
MCS161015

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Majid Iqbal Khan	COMSATS University Islamabad
(b)	Internal Examiner	Dr. Muhammad Arshad Islam	CUST, Islamabad
(c)	Supervisor	Dr. Muhammad Tanvir Afzal	CUST, Islamabad

Dr. Muhammad Tanvir Afzal
Thesis Supervisor
November, 2018

Dr. Nayyer Masood
Head
Dept. of Computer Science
November, 2018

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
November, 2018

Author's Declaration

I, **Muhammad Aadil-ur-Rehman** hereby state that my MS thesis titled **"Determination of Feature Contribution Score and Feature Ranking in Author Name Disambiguation"** is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Muhammad Aadil-ur-Rehman)

Registration No: MCS161015

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled **"Determination of Feature Contribution Score and Feature Ranking in Author Name Disambiguation"** is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Muhammad Aadil-ur-Rehman)

Registration No: MCS161015

Acknowledgements

First of all, I would like to express my gratitude to ALLAH Almighty. Secondly, I would like to thank to my supervisor Dr. Muhammad Tanvir Afzal for the guidance and encouragement at every stage of the research work. He is a true blessing. Last but not the least; I would like to thank my family for their endless support and prayers. I would like to express my thanks to a friend Mr. Rizwan, his discussions and critic inputs helped me a lot in accomplishing this task. Last but not least, I would like to thank my friends who helped me in this task directly or indirectly.

Abstract

Authors showcase their scientific contributions by publishing papers in journals and conferences, workshops. These publications are indexed in digital repositories like, DBLP, MEDLINE, CiteSeer, arXiv, MAS, BDBComp, Google Scholar etc.. This publications and citations data is used to compute different metrics for author ranking i.e., DS-index, H-index, G-index, R-index etc.. This information is very useful for making important decisions such as granting funding and research awards, impact factor calculation, journal ranking, expert finding etc. Citations are indexed by author names. Authors usually use their full name or their abbreviations for their first or last names in their publications. Due to natural limitation of names, ambiguities like polysem and synonym occurs i.e., the same author may publish using different name variants, or different authors may publish using the same name. The problem of assigning true authors to their own citations and publications is known as author name disambiguation.

A Myraid of efforts have been done in this domain. Broad categories of these methods are Machine learning based methods including supervised, unsupervised, and semi-supervised methods , heuristic based methods and graph based methods.

After critical analysis of literature in the domain, this research has identified the following facts and research gaps: (1) The contemporary approaches have utilized the following features for author name disambiguation: (a) Title, (b) Co-authors, (c) Venues. (2) Different researchers have used either one of the above feature or have combined the above features in different ways. (3) Previous approaches have used the following classifiers: (a) Decision Tree, (b) Naïve Bayes, (c) Random Forest, (d) Bagging None of the previous approach has conclusively outlined the contribution score of each feature and have not comprehensively evaluated and explained that for example, when you have only Titles of the research papers, in that situation, which classifier should be used. The above research gap has led us to explore the answers of the following research questions: (RQ1) What is the contribution score of each feature, how they can be ranked based on their impact.

(RQ2) Which classifier should be used to get maximum accuracy based on the feature set one has at hand?

In this study, We derived contribution score calculation formula. Four well known machine learning algorithms are used i.e., decision tree, naive bayes, random forest and bagging. We comprehensively evaluated these algorithms over three most widely used datasets for author name disambiguation i.e., DBLP, Kisti, and BDB Comp. Average F-Measure is used as evaluation metric for contribution score calculation.

The source code and datasets is made publically available at:

https://github.com/maadilrehman/contribution-score-in-AND, to help the interested research community in this domain.

Contents

Author's Declaration iv

Plagiarism Undertaking v

Acknowledgements vi

Abstract vii

List of Figures xii

List of Tables xiii

Abbreviations xiv

Symbols xv

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Research Gap . . . 2
- 1.3 Problem Statement . . . 3
- 1.4 Purpose . . . 3
- 1.5 Scope . . . 3
- 1.6 Applicaitons of Proposed Solution . . . 4

2 Literature Review 5
- 2.1 Introduction . . . 5
- 2.2 Challenges in Author Name Disambiguation . . . 7
  - 2.2.1 Polyseme (Homonym) Problem in Author Name Disambiguation . . . 7
  - 2.2.2 Synonym Problem in Author Name Disambiguation . . . 8
- 2.3 Methods for Author Name Disambiguation . . . 8
  - 2.3.1 Supervised Machine Learning Algorithms . . . 9
  - 2.3.2 Un-supervised Machine Learning Algorithms . . . 11
  - 2.3.3 Semi-Supervised Machine Learning Algorithms . . . 14
  - 2.3.4 Heuristic Based Algorithms . . . 17

2.4 Critical Analysis . . . 18
2.5 Summary . . . 19

3 Proposed Methodology **21**
3.1 Introduction . . . 21
3.2 Datasets . . . 21
3.2.1 DBLP Extracted . . . 22
3.2.2 KISTI . . . 23
3.2.3 BDBComp . . . 24
3.3 Features . . . 25
3.4 Preprocessing . . . 26
3.5 Techniques . . . 27
3.5.1 Naïve Bayes Classification . . . 27
3.5.2 Decision Tree . . . 28
3.5.3 Random Forest . . . 29
3.5.4 Bagging - A Voting Based Ensemble Learner . . . 29
3.6 Evaluation Matrices . . . 30
3.6.1 Precision . . . 30
3.6.2 Recall . . . 30
3.6.3 F-measure . . . 31
3.6.4 Finding Contribution of Evidences . . . 31
3.7 Summary . . . 33

4 Experiments and Results **35**
4.1 Introduction . . . 35
4.2 Preprocessing . . . 35
4.3 Evaluation . . . 36
4.4 Algorithms Evaluation . . . 36
4.4.1 Individual Feature Evaluation . . . 37
4.4.1.1 DBLP . . . 37
4.4.1.2 Kisti . . . 37
4.4.1.3 BDBComp . . . 38
4.4.2 Group-wise Feature Results . . . 39
4.4.2.1 DBLP . . . 39
4.4.2.2 Kisti . . . 40
4.4.2.3 BDBComp . . . 40
4.5 Contribution Score Evaluation . . . 42
4.6 Individual Feature Contribution Score . . . 42
4.6.0.1 DBLP . . . 42
4.6.0.2 Kisti . . . 43
4.6.0.3 BDBComp . . . 44
4.7 Feature Groups Contribution Score . . . 45
4.7.0.1 DBLP . . . 45
4.7.0.2 Kisti . . . 46

4.7.0.3 BDBComp 47
4.8 Summary 49

5 Conclusion and Future Work **50**
5.1 Conclusion 50
5.2 Future Work 52

Bibliography **53**

List of Figures

3.1 Methodology Diagram. 22

4.1 Evidences-based datasets 36
4.2 Contribution Score of individual features for DBLP Dataset 42
4.3 Contribution Score of individual features for Kisti Dataset 43
4.4 Contribution Score of individual features for BDBComp Dataset . . 45
4.5 Contribution Score of individual features for DBLP Dataset 46
4.6 Contribution Score of individual features for Kisti Dataset 47
4.7 Contribution Score of individual features for BDBComp Dataset . . 48

List of Tables

2.1 Sample Ambiguous group 6
2.2 Polysem Example 8
2.3 Synonym Example 8
2.4 Comparison of studies for Author Name Disambiguation 19

3.1 Ambigous Author Groups in Selected DBLP Collection 23
3.2 Ambigous Author Groups in Selected KISTI Collection 24
3.3 Ambigous Author Groups in Selected BDBComp Collection 25

4.1 Average F-measure for DBLP Dataset of individual features 37
4.2 Average F-Measure Results of Kisti Dataset for individual features . 38
4.3 Average F-Measure of BDBComp Dataset for Individual features . 39
4.4 Average F-measure for DBLP Dataset for Groups 40
4.5 Group wise Average F-Measure Results of Kisti Dataset 40
4.6 Average F-Measure of BDBComp Dataset for Groups 41
4.7 Average F-Measure for Title-Venue-CoAuthors for all datasets . . . 41

Abbreviations

AND	Author Name Disambiguation
MAS	Microsoft Academic Search
DL	Digital Library
CUST	Capital University of Science and Technology
ML	Machine Learning
NB	Naïve Bayes
SVM	Support Vector Machine
RF	Random Forest
WEKA	Waikato Environment for Knowledge Analysis

Symbols

IG	Information Gain
P	Precision
R	Recall
F	F-measure
G_i	Set of feature Combinations containing feature i
U_i	Set of feature Combinations excluding i
I_i	Contribution Score of feature i

Chapter 1

Introduction

1.1 Background

Authors make scientific contributions and publish papers in: journals, conferences, books, and workshops. These publications are then indexed in different scientific data management systems like DBLP, MEDLINE, CiteSeer, arXiv, MAS, Google Scholar, BDBComp etc.

Authors are ranked in their disciplines based on their contributions. Their publications, citations or some author-level metrices such as DS-index, H-Index, G-Index, R-index, AR-index etc. are used as parameters for their ranking. This information helps in making important decisions such as granting funding, research awards, impact factor calculation, and journal ranking. [1]

Citations are indexed by author names in online academic systems Authors usually use their full name or their abbreviations for their first or last names in their publications, this often generates same abbreviated name for different authors like Rizwan Yasin as R.Yasin and Ramzan Yasin as R. Yasin or same author with different variants of its name, or different authors with same names. This led not only the problem for an automatic system to detect which abbreviation belongs to whom but also is quite difficult for manual processing. This problem of assigning true author to a citation is formally known as author name disambiguation. The

variants of the same name are referred as synonyms whereas, the different names sharing the same representation is usually termed as polyseme.

In automatic systems, a paper written by an author maybe associated to a different author because of their name ambiguities. Author name disambiguation is a long-standing issue, there is a myriad of solutions proposed in the literature which can be classified into five broad categories based on their internal algorithmic techniques, i.e., supervised machine learning methods , unsupervised machine learning methods, semi-supervised machine learning, heuristics-based methods and graph-based methods [2] .

The various proposed techniques for author name disambiguation uses different features of publications which include title, Co-authors, Publication Venues, affiliations, keywords, abstracts, publication years, publication age, topic models etc. [3–5]. These features are usually not available at once, therefore, working on larger scales is unrealistic for the researchers. In a citation entry usually one can only find, title of publication, co-author names, and venues, therefore, most of the studies try to focus on these features only.

1.2 Research Gap

After critical analysis of the literature in the domain, this research has identified the following facts and research

1. The contemporary approaches have utilized the following features for author name disambiguation: (a) Title, (b) Co-authors, (c) Venues.
2. Different researchers have used either one of the above feature or have combined the above features in different ways
3. Previous approaches have used the following classifiers: (a) Decision Tree (b) Naïve Bayes, (c) Random Forests, (d) Bagging - A voting based ensemble classifier

4. The state-of-the-art approaches have used different classifiers for author name disambiguation.

None of the previous approach has conclusively outlined the contribution score of each feature and have not comprehensively evaluated and explained that for example, when you have only Titles of the research papers, in that situation, which classifier should be used etc.

1.3 Problem Statement

The above research gap has led us to explore the answers of the following research questions: (RQ1) What is the contribution score of each feature, how they can be ranked based on their impact. (RQ2) Which classifier should be used to get maximum accuracy based on the feature set one has at hand?

1.4 Purpose

The purpose of this thesis is to determine the contribution of each individual evidence and combinations of evidences in author name disambiguation task.

1.5 Scope

The scope of this thesis is determining the contribution of evidences in by using four well known classification algorithms. Contribution will be determined for each individual evidence as well as for all combination of evidences. The results of this study will be immensely important for the author name disambiguation methods in selection of evidences for classification of authors. The methodology of this study will help in determining the contribution of evidences in classification for other evidences as well as the methodology can be applied to other problem domains as well.

1.6 Applicaitons of Proposed Solution

This research can assist in various fields such as:

1. Author Name Disambiguation
2. Authors Ranking
3. Determining Feature Contribution in Classifications
4. Bibliographic studies
5. Digital Libraries

Chapter 2

Literature Review

2.1 Introduction

As we stated in the chapter 1, author name disambiguation in digital library is the problem of assigning the true authors to their citations [2, 6–11]. Author name disambiguation is a serious concern not only for the digital libraries but it also affects the accuracy in other domains i.e., web search, document retrieval, information fusion, author ranking, and expert finding [1, 2, 10, 12–14]. This chapter presents the literature review for this study. To understand the problem, an example is derived from DBLP dataset.

In Table 2.1, Three citations c1, c2, c3 are shown, each citation has its author names identified by id "ri". Each ri refers to an author. The citation records shown in Table 1.1. Author r7 and r8 are different authors sharing the same name, where r7 refers to "Micheal L Miller" from "Washington University St. Louis, Missouri" and r8 refers to "Mark S Miller" from "Erights.com, USA", which represents the polysem problem. Authors names r7 and r4 are different but they both belongs same author "Micheal L Miller", which represents the synonym problem. The removal of such ambiguities is known as Author Name Disambiguation (hereinafter referred to as AND).

TABLE 2.1: Sample Ambiguous group

Citation Id	**Citation Details**
C1	(r1)J A O'Sullivan, (r2) MD DeVore, (r3)V Kedia, (r4) Michael Miller "SAR ATR performance using a conditionally Gaussian model. "IEEE Transactions on Aerospace and Electronic Systems 37, no. 1 2001. 91-108.
C2	(r5)P Dupuis, (r6) U Grenander, (r7) M Miller "Variational problems on flows of diffeomorphisms for image matching." Quarterly of applied mathematics. 1998, pp.587-600.
C3	(r8) M Miller, (r9) D Krieger, (r10) N Hardy, C Hibbert "An automated auction in ATM network bandwidth."Market-Based Control: A paradigm for distributed resource allocation. 1996. 96-125.

More Formally, Let C = c1, c2, ..., ck be a set of citation records, let R =rA1, rA2 be the set of real authors. AND problem can be defined as the problem of Assigning each ci to its real author rAj.

There are a number of methods proposed in the literature for author name disambiguation [2] though there have been significant advancements but there is a lot of space for further improvements. Lack of neat and clean data significantly effects the results [15–18]. The number of ambiguous authors and clusters is not known in majority of unsupervised author name disambiguation methods [17–20]. With the increase in the number of ambiguous authors scaling some of the techniques is not possible [17, 21–24]. Some techniques use Web resources i.e., Social Media Profiles or Personal Home Pages to extract other features or user feed-backs for the process of disambiguation [24–28].

Different evidences of publications such as title words, co-authors, affiliations, keywords, abstract words, publication years and references are used for the solution of author name disambiguation [4, 5]. Due to lack of availability of all these attributes at once, working on larger scales is unrealistic for the researchers. Therefore, an increased amount of publication resulting into huge digital libraries has been a significant factor for the author name disambiguation in recent times [25].

A comprehensive literature review is carried out and critically evaluated for this

research. The evaluation was performed using the following parameters: Methodology adopted, dataset(s) used, features used, results and limitations of the study.

Many efforts have been done in this regard, the AND methods are classified into four broad categories according to the type of approaches they are using i.e., supervised machine learning methods, unsupervised machine learning methods, semi-supervised machine learning heuristics based methods [2].

In this chapter, we will review the AND challenges in section 2.1. We will examine AND methods in section 2.2. Critical analysis of the literature reviewed will be discussed in section 2.3. Finally, we will summarize this chapter in the section 2.4.

2.2 Challenges in Author Name Disambiguation

Many researchers have conducted the research regarding the author name disambiguation problem. There are two sub-challenges in author name disambiguation problem: polyseme and synonym, these challenges are defined in section 2.2.1 and section 2.2.2 respectively.

2.2.1 Polyseme (Homonym) Problem in Author Name Disambiguation

The first challenge in author name disambiguation is polyseme [10] or homonym problem, which means that multiple authors have the same name. This problem is also known as mixed citation problem [29].Two citations are shown in Table 2.2, The author r3 and r4 have the same name in citations but they are actually two different authors. r3 refers to real author "Micheal L Miller" whereas r4 refers to real author "Mark S Miller".

TABLE 2.2: Polysem Example

(r1)P Dupuis, (r2) U Grenander, (r3) M Miller "Variational problems on flows of diffeomorphisms for image matching." Quarterly of applied mathematics. 1998, pp.587-600.
(r4) M Miller,(r5) D Krieger, (r6) N Hardy, C Hibbert "An automated auction in ATM network bandwidth." Market-Based Control: A paradigm for distributed resource allocation. 1996. 96-125.

TABLE 2.3: Synonym Example

(r1)J A O'Sullivan, (r2) MD DeVore, (r3) V Kedia, (r4) Michael Miller "SAR ATR performance using a conditionally Gaussian model." IEEE Transactions on Aerospace and Electronic Systems 37, no. 1 2001. 91-108.
(r5)P Dupuis, (r6) U Grenander, (r7) M Miller "Variational problems on flows of diffeomorphisms for image matching." Quarterly of applied mathematics. 1998, pp.587-600.

2.2.2 Synonym Problem in Author Name Disambiguation

The second challenge is synonyms, which means that the same individual has different name variants. The problem is also known as split citation by [29]. Table 2.3 shows two citations, here r4 and r7 have different author name but in reality, both r4 and r7 refers to the same real author "Micheal L Miller".

2.3 Methods for Author Name Disambiguation

As said before, one way to organize the several existing author name disambiguation methods is according to the type of approach they employ. The two approaches are author grouping and author assignment. There are some methods that are using hybrid approach i.e., using both author grouping and author assignment methods. Such all methods have been critically discussed in the following sub-section.

2.3.1 Supervised Machine Learning Algorithms

In Supervised machine learning algorithms, manually labeled training data is inputted into the classifier. The data comprises of a pair of input feature vector Ai and the labeled output class Bi making the form (Ai,Bi). The training data is mapped to get the correct output class value. The training data helps in training and validation of a classification model. 'K' fold cross-validation technique is one of the techniques that are used for validation. 'k' can be any number between 2 and N. Prediction of unseen output data is made through this trained model. For example, co-authors, title words, year of publications and venue information can be the input data and true author name class can be the output.

A boosted tree classification method was proposed by [21]. There are four steps of boosted tree classification method. Firstly, initial and last names are matched along with the affiliations and secondly, similarity scores are calculated for six publication features. In the third step, false rate is calculated. A data set is manually created containing 4253 citations of 100 authors. The boosted tree classifier is applied on this data set in the last step. Manual checking is required due to higher false rates which cannot be classified by the classifier itself.

A deep neutral network-based approach proposed by [17] could automatically process on any data set and disambiguate the author names. There are two components involved to this solution. String matching was used by the authors to compute the data representations of the input data in the first component. In the second component these features are used in training to disambiguate the author name. Probabilities are computed on the forwarded feed from other components in order to find the similarities of the author names of the data set. For the purpose of generalization, multi column deep neutral network technique was used.

Two extreme learning machine-learning-based algorithms [15] named One classifier for each name (OCEN) and One classifier for all names (OCAN). OCEAN uses author names, title words of papers and title words for venues as features. Using some attributes, the classifiers are trained so that when an unseen paper is given

to the classifier it identifies the author. Principal component analysis is used to minimize the dimensionality and later in the end extreme learning machine method is used for optimization.

OCAN is not specific with any particular name. The classifier is trained to predict the similarities of different entities. A pair is used to abstract from concrete names. The classifier so disambiguate all names. Similarities between author names, title name words and venue title words are formulated. Additional information on the relationships regarding similar author names are gathered through enhanced feature extraction. In the end extreme learning machine algorithms are applied to find solution. The performance is compare by support vector machine classifier to measure the performance generalization.

A two-stage filtering process was used by [30]. Logistics regression was used with a discrimination functions was predicted the true authors and homonyms. The data set consisted of Web of Science was used. The filtering process retrieved 629000 papers. Initially, all those papers with no citation relationship between retrieved and source papers were removed also along with those which had low address similarities. Secondly, from manual judgments the discrimination function was determined for logistic regression. The features included co-authors, address similarities, title words similarities and relationship of citations between source and retrieved papers. This technique is not good for the papers whose subject fields and affiliation addresses are similar or vary a little.

For the solution of Vietnamese author name ambiguity, [31] proposed five supervised machine learning algorithms which were Random Forest, Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Decision Tree (C4.5) and naive Bayes (NB). Levenshtein similarity was used. For assistance in training classifiers, a set of features from publication data set was proposed. Very clear, specific, concise and relevant data set is a must for the training of the model.

2.3.2 Un-supervised Machine Learning Algorithms

In un-supervised machine learning algorithms, unlabeled input data is used to find intrinsic patterns in order to determine the correct output values. With this approach, it is assumed that there exists such pattern structure which hold certain similarities and can be utilized to determine the correct outputs. It is a harder task to identify similarities between patterns. These algorithms use some predefined similarity forms and functions which are used to identify that cluster of ambiguous author names.

A self-training method SAND is proposed by [16]. This method is divided into author grouping, cluster selection and author assignment. In the first step, clusters of citation records are automatically generated, Pure cluster are obtained by exploiting relations between co-authors from the records. If there is at least one similar co-author name or at least two have not common last names, then SAND consider that the citation record and cluster share a co-author. Secondly, in cluster selection step, the cluster which are not similar and have larger number of citations are selected and the citation records along with the corresponding author labels are introduced into the training data. In the third step, on basis of a lazy associative classifier, sets of training data are used for production of a disambiguation factor to predict the correct author from the unselected clusters. Representative clusters in the training data are detected by SAND in included into the training data. Exploitation of reliable predictions helps increasing the coverage of training data.

An incremental author name disambiguation (INDi) was proposed by [32]. The authors with new citation records are determined by INDi when they got added into a DL. It is not applied on the DL at to save manual corrections that are done before. Special heuristics are used by INDi to determine if there is a link between the author names of new citations to the pre-existing authors in the DL or the ones with no citation records. Instead of assigning the doubtful record to an existing author with a probability of error. The heuristics are used to disambiguate new citation records by prioritizing the assignment of such records to the correct

authors. The new citation records are disambiguated by reaching to an existing author with similar author name in the DL i.e. at least one common co-author, similar work and publication venue titles. It does not perform co-author checks and raises similarity thresholds for publication venue and work title in cases where the new citation records do not include co-authors or all existing records in a group of an existing similar author do not include an co-authors. the citation is considered to belong to a new author when the checking procedure fails.

An algorithm for author name disambiguation that used Dempster-Shafer theory (DST) and Shannon entropy(SE) was proposed by [20]. Web correlations and correlation similarities were calculated for features i.e., affiliation, publication venue, content, co-authors and citations. Later, these features were combines using DST and SE, and then belief and plausibility were calculated using the combined information. A matrix of pairwise correlation of papers got generated. Each entry of this matrix was then linked to the belief and plausibility function. In the end, three different conditions namely; preset number of cluster, number of available evidences and distance between clusters were used to apply the DST-based hierarchical agglomerative clustering for author name disambiguation.

An algorithm was proposed by [33], that is using cognitive maps of psychology and structural evidence of network analysis-based knowledge homogeneity scores to recognize author name ambiguity in bibliography. The basic assumption here was that all the authors have a specific set of knowledge base at a specific time. Therefore, in any given time, two different authors if have same knowledge base, are considered to be the same and authors of same name but distinctive knowledge base are considered to be different authors. Since complete similarity of authors is rare to find, approximate structural equivalent (ASE) is used in a way that the authors within a cluster have similarities within and are different from the ones outside the cluster. These authors are considered same if there is similarity in the family name and first initial. Knowledge homogeneity similarity (KSE) score (using the sum of shared references, forward citations and minimum number of references in two articles), is used to find the ASE. After the construction of KHS matrix, groups are distinguished by doing hierarchical clustering with single

linkage. Performance of this method is compromised if the articles lack references to the citations.

Markov random fields are formalized to resolve the author name disambiguation problem using a unified probabilistic framework proposed by [21]. The data is consistent on local attributes and relationships. An algorithm is suggested that estimates the unknown ambiguous authors's count and features. In this method it is assumed that an author can be identified if having similar content and similar relationship. This technique uses the advantage of interdependencies between paper assignments. Firstly, features i.e., Title word, venue name, publications' year. Abstract, authors and references, are assigned to every paper which is retrieved from the online digital libraries. Five relationships were defined, and weights are given to them which are unknown. The content-based information and structure-based information is then transformed into the hidden Markov random fields (HRMF). An objective function is defined as the maximum a posteriori configuration of the HMRF. The true number of authors is estimated by using Bayesian information criterion. An algorithm is devised to find unknown parameters. First initialization in assignment of random values to unknown parameters is done by this algorithm. After assignment, these values are updated to each function value in order to achieve optimization.

A technique proposed by [34] for resolving mixed and split citations by first creating the clusters based on disciplines and dividing them to small clusters using co-authors feature, then using the remaining features to merge the clusters given their distance is less than a defined threshold. User feedback is required for the retrieved clusters to purify them.

An algorithm that calculate the h-index of the authors along with the disambiguation or author names was proposed by [35]. The number of shared co-authors, self-citations, common references and the number of papers citing both publications are used to calculate the pairwise similarities between all publications. A link between those publications having greater similarity than a set level is constructed while calculating the similarities of the pairs. A distinct author is found

by knowing connected components forming clusters. New similarities are calculated between these clusters. Again, a link is identified and determined if its above a certain set threshold. Later, all these clusters are combined which are therefore set of papers by unique authors. The disambiguated authors are optimized and validated by re-calculating their h-index. H-index feature is vital in this method.

A technique that used a three-step clustering method for author name disambiguation called "Fast Multiple Clustering" proposed by [36]. With the help of co-authors, the cluster of such authors are extracted in which different relations i.e., papers related to author or paper related to paper are found. These related papers are then clustered into various clusters. Using similarities of title words, bigger clusters are formed. Venue information used as feature to cluster the publications of the authors, who usually publish in the same venue, but the titles differ.

A Multiple layer's based name disambiguation framework was introduced by [37]. It classified the AND subproblems as NMA and NSA. NMA means, don't mix records of two different authors, while NSA means, don't split the records of same author. A multi-layer approach is adopted for clustering, a set of clusters created in each layer were given to next layer as input. Firstly, email-based clustering is performed, then co-authorship-based clustering is performed on the previously created clusters. For co-authorship-based clustering Erdos Numbers are calculated and package-merge-based clustering is performed. In the third layer, topic-based clustering is performed using dynamic clustering algorithm. Emails are being extracted from the web links or PDF files of full texts but as the layers are independent of one another, so email layer can be easily removed.

2.3.3 Semi-Supervised Machine Learning Algorithms

In semi-supervised machine learning algorithms, a hybrid approach is used. Both labeled and unlabeled training input data is used to achieve higher accuracy. It is assumed that both unlabeled and labeled data are likely to have same label due

to much similarities in their patterns and structures. A lot of success is achieved in development of algorithms for AND using semi-supervised learning approach. [34, 38, 39].

A hybrid name disambiguation framework proposed by [40]. This framework used co-authors and web page genre information. The main objectives were to identify the web page and re-cluster the model. The first step, the it is checked if the returned web pages belong to the authors or not. The belonged web pages are the disambiguated, rest are disambiguated using co-author information. The records that are left behind are then sent to the re-clustering model. The citation records are used to build a graph G in which the citation record is presented by vertex and author relationships of same domains are presented by the edges. Relations are assumed if there is enough evidence of links between two vertices. Here multi-dimensional scaling algorithm is used to turn the graph into a similarity matrix and detect the homogeneity among the objects. Two-dimensional matrices of co-authors and topics are constructed. Eucilidean distance is used for similarity calculation between vertices. The citations are considered to be from the same author if the distance between the citation is lesser than the set threshold. There is limitation to this approach if the citations on one personal page of an author are different than the citations on another personal page of the same author because here the authors are considered as two distinct authors.

A semi-supervised two stage method was presented by [18] for disambiguation of authors in DLs. Labeled training data was created automatically by using citation-based rules in the first stage. In the second sage of agglomerative clustering, the initial clusters are used to find the similarity matrices. Both old and new features were used to measure the similarities of publications. The data set of Thomson Reuters Web of knowledge is used for evaluation of the model.

A semi-supervised approach using Microsoft academic research data was proposed by [38]. Initially data is pre-processed and co-author based bibliographic network is constructed. The community detection algorithm is applied on this data and the uncertainty in data gets handled by support vector machine along with

other machine learning algorithms. A 0.9877 mean F-score was achieved Microsoft academic search data set provided by KDD cup 2013 after merging the results.

An ethnicity sensitive method proposed by [39]. This method had three parts. Initially similar author signatures are blocked on the basis of phonetics. Then to exploit more sensitive information, supervised machine learning linkage function is used. The categories of ethnic groups are used to divide the authors. these categories are white, black, American Indian or Alaskan native, Chinese, Japanese, Asian or pacific islanders and others. Origin of the authors is predicted on the basis of probability measured for a pair of names. The difference between the two pairs of linkage function us then used for hierarchical agglomerative clustering.

Web co-relations and author co-relation-based approach in order to estimate the similarities between publications for author names ambiguity was proposed by [40]. Pairwise similarity metrics which uses modified sigmoid function, cosine metric and name popularity metrics, is used to measure both web co-relation and author co-relation. It was under assumption that the citations on a webpage are related to the that author and citations with a rarer author belongs to that same author.

To address the issue of discarded null data fields and its implications on F-measure, recall and precision, an algorithm proposed in [14]. Tan Mao similarity coefficient was used on all data fields including title and abstract words, first initials and last names of co-authors, whole strings of cited references, normalized author keywords, normalized indexed keywords, normalized research addresses and venue. To get increased precision and recall of returned records, average author contribution and age difference between publications were included. Weights to all unknown parameters are obtained by applying the logistics regression. In the Blondel community detection algorithm is applied in order to find the author names.

A model that can solve for both homonyms and synonyms using semi-supervised machine learning algorithms was proposed by [41]. Training data is not required and semi-supervised approach is used for author names. Fusion of attributes us done by applying the multi-aspect similarity indicator and support vector machine.

In the end, to increase the performance of disambiguation, a self-taught method is presented

2.3.4 Heuristic Based Algorithms

Heuristic based algorithms are used when quick and precise results are required. These results may not be as precise to be perfect but are somewhat close to perfect. These algorithms are used with the information at hand and reaches out to the most precise solution available.

A heuristic based hierarchical clustering (HHC) was proposed by [42]. It was based on two assumptions: (a) two authors having similar names and sharing a common co-author are very rarely two different persons and (b) for a considerable tenure of the career, an author publishes in same domain and venue. This technique is staged in two steps, first step groups the records based on heuristic (a) similar author and common co-author-based clusters are generated in this step. In second step initially created clusters are merged using similarity of publication venue or work. Merger step continues until no more clusters can be merged.

An incremental unsupervised name disambiguation (INDi) approach was proposed by [32]. In INDi clustering in performed by computing similarity among bibliographic recrods.Custom defined heuristics are applied to check weather a record belongs to new cluster or not. Experiments were performed on a BDBComp dataset. It does not consider the heuristic for new authors.

A name matching framework for author name disambiguation for Microsoft Academic Search data set is proposed by [43]. This method consists of six stages in total. In first stage Chinese or non-Chinese groups are separated by using Chinese name dictionaries. Second stage preprocess citations. In the third stage, the technique uses blocking strategy to create blocks of similar author names. Blocking is based on dictionary of terms in author name. High false positive and low recall strategy is adopted in this stage. Then the duplicates are identified in fourth stage.

Authors are linked to their identifiers in fifth stage. Merger is performed in sixth step by filtering based on background information.

2.4 Critical Analysis

After going through the comprehensive analysis of state of the art approaches in the domain, we found that techniques for author name disambiguation (AND) are using different features, datasets and techniques. This section defines the critical analysis of the reviewed techniques. A brief overview of reviewed literature is presented in Table 2.4. The parameters for the literature review were adopted methodology, features, datasets, limitations of the study and the evaluation metrices used in the study.

In Table 2.4, it can be observed that the techniques are based on different features. These features can be classified into three different categories i.e., Citation meta-data based features, Publication Content based features, and Features extracted from external sources such as Social Media Profiles and Personal Home pages etc.

Citation meta-data based features are freely available in digital libraries while publication content is not always freely available. Features extracted from external sources require overhead of crawling and parsing the data. External sources are not always available for each author and sometimes it requires manual annotation to check the authenticity of the external sources. Even for the freely available citation meta-data based features, different crawling techniques are used to parse the features. These techniques can not always parse all of the features. So the author name disambiguation technique is applied on the feature set at hand.

In the literature, The techniques are using different features but none of the techniques has conclusively outlined the contribution score of each feature and have not comprehensively evaluated and explained that for example, when you have only Titles of the research papers, in that situation, which classifier should be used etc.

TABLE 2.4: Comparison of studies for Author Name Disambiguation

Ref	Methodology	Dataset	Features	Results	Limitations
[44]	Heuristics	Custom	AuthorName, PublicationTitle, AuthorArea, AuthorAffiliation, VenueTitle, PublicationDate, Co-Author	F1= 0.91	1) using Content
[45]	Heuristics	MEDLINE	AuthorName, PublicationTitle, PublicationKeywords, PublicationContent, AuthorAffiliation, VenueTitle,Co-Author	P=99.51 R=99.64	1) Heuristics without Feature ranking
[46]	re-clustering model	DBLP	AuthorName, PublicationDate, Co-Author, PersonalHomepage	F1= 0.9	1) crawling Overhead 2) Personal Home page is not available
[14]	Community Detection	Custom	AuthorName, AuthorAddress, AverageAuthorContribution, PublicationTitle, PublicationAbstract, PublicationKeywords, PublicationContent, AuthorArea, AuthorAffiliation, VenueTitle, PublicationDate, Co-Author	P=90% R=85% F=0.8	1) Area computation overhead
[47]	agglomerative Clustering, h-index distribution Model	Kisti,DBLP	AuthorName, PublicationTitle, PublicationContent	P=88% R=87%	1) Citation network generation overhead
[40]	classification using topic Model	Kisti,DBLP	AuthorName, PublicationTitle, PublicationKeywords, AuthorArea	F1=91%	1) Content required
[31]	Random Forest, SVM, K-NN, Decision Trees, Naīve Bayes	ACM Vietnamese DS	AuthorName, PublicationTitle, PublicationDate, Co-Author	Accuracy=89%	1) No feature ranking

2.5 Summary

In this chapter, the literature review for author name disambiguation is presented. Many techniques have been proposed. These techniques are classified into different categories: Supervised Machine Learning Algorithms, Unsupervised Machine

Learning Algorithms, Semi-Supervised ML Algorithms, Heuristics Based Algorithms. All author name disambiguation techniques depend upon the availability of features and data used for training and testing. In this research, we are interested in finding the contribution score of features in author name disambiguation and ranking of features. The next chapter defines the adopted methodology for determining feature contribution.

Chapter 3

Proposed Methodology

3.1 Introduction

The comprehensive exploration of state-of-art techniques presented in chapter 2 shows that all the performance and efficiency of techniques used in AND are subjected to the quality of training and testing data as well as on the evidences used. In this study, we are finding the contribution of the evidences used for AND. This section explains methodological steps that will be followed to evaluate the evidences used for author name disambiguation. The block diagram of the proposed model is shown in the Figure 3.1.The detailed description of each module of figure 3.1 is given the next sections.

3.2 Datasets

Benchmark datasets are required for evaluation of features and methodology. In AND many datasets have been used i.e., DBLP, BDBComp, Kisti, MAS, INSPIRE, WOS etc., [2, 10]. Some techniques defined self-curated datasets [15]. Some techniques use SyGAR [48], a synthetically dataset generator for AND. We used 3 datasets for this research DBLP Extracted, Kisti, and BDBComp.

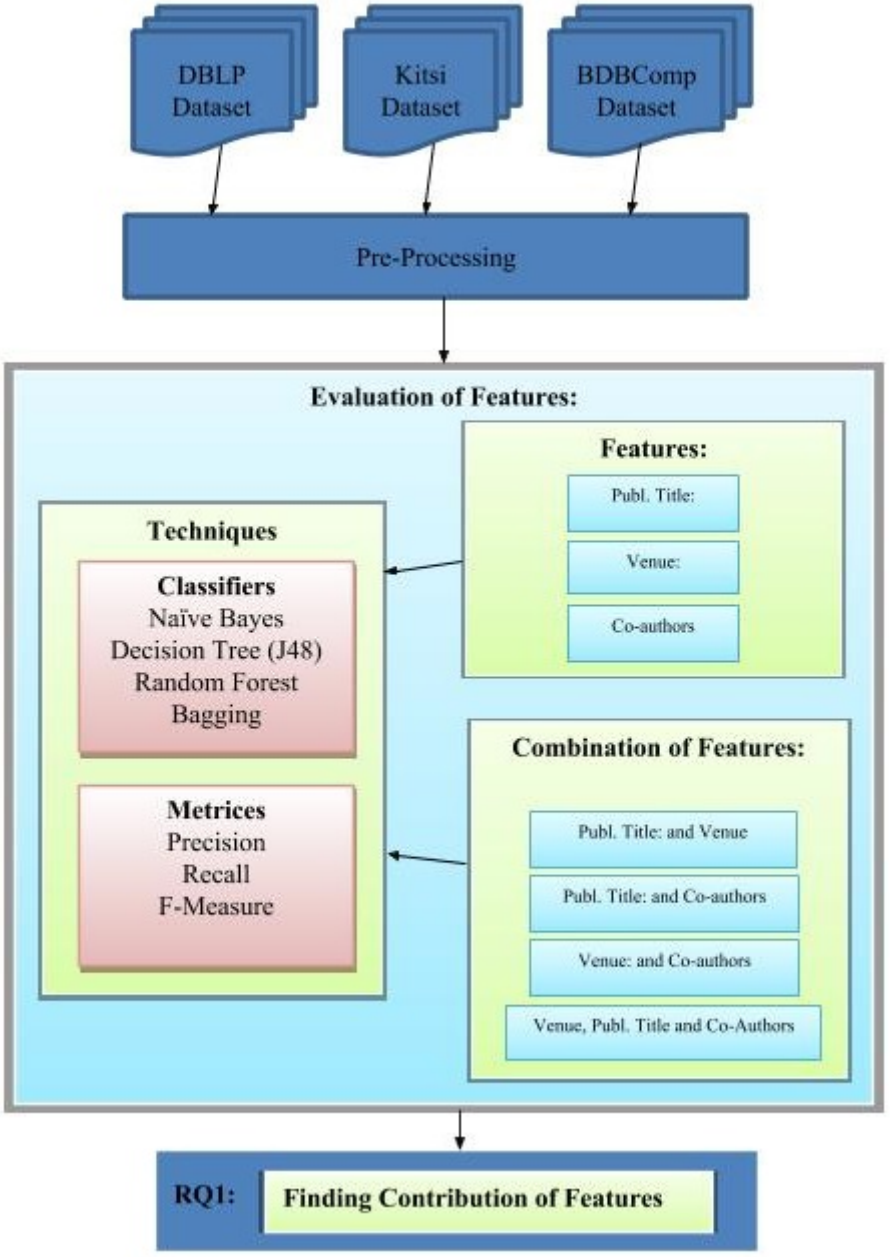

FIGURE 3.1: Methodology Diagram.

3.2.1 DBLP Extracted

DBLP is most widely used dataset in author name disambiguation techniques as seen in the literature review. The DBLP Team manually collects publications in computerscience fields and disambiguate them. There are 4 billion publications populated in DBLP till December 2017. Its dataset is publicly available online in the form of XML files. Many techniques have used a subset of DBLP dataset

TABLE 3.1: Ambigous Author Groups in Selected DBLP Collection

Author Group	Number of Authors	Number of Citations
A. Gupta	563	26
C. Chen	791	60
D. Jhonson	365	15
J. Lee	1397	100
J. Smith	900	29
K. Tanaka	280	10
M. Jones	260	13
M. Miller	411	12
S. Lee	1347	86
Y. Chen	1261	71

created by [19], with slight variations. We preferred to use a variation of same dataset used by [49]. It contains 4287 publications and 220 unique authors.

Number of Authors and Number of Citations in each ambiguous group are defined in Table 3.1.

3.2.2 KISTI

Kisti dataset was built by the Korean Institute of Science and Technology Information [50] specifically for author name disambiguation. It comprises the citation records from the top 1000 most frequent author names from late-2007 DBLP. A reference was built for each author name in each citation record. The manual disambiguation relied on Google to retrieval authors personal publication pages. Manual inspection of the first retrieved web pages identified the correct personal publication page. This collection has 37,613 citation records, 881 groups of same-name persons and 6,921 authors.

TABLE 3.2: Ambigous Author Groups in Selected KISTI Collection

Author Group	Number of Authors	Number of Citations
yzhang	208	56
ywang	257	62
yliu	312	52
ychen	301	71
xli	304	54
sjajodia	218	2
jwang	325	59
jlee	203	66
jchen	252	61
hwang	268	36

Number of Authors and Number of Citations in each ambiguous group are defined in Table 3.2.

3.2.3 BDBComp

This collection was created by us based on the Brazilian Digital Library of Computing. It comprises 363 records associated with 184 distinct authors: about 2 records by author. Despite its small size, this collection is difficult to disambiguate, with many authors having only one or two citation records. It contains the 10 largest ambiguous groups in this repository considering the period between 1987–2007.

Number of Authors and Number of Citations in each ambiguous group are defined in Table 3.3.

TABLE 3.3: Ambigous Author Groups in Selected BDBComp Collection

Author Group	Number of Authors	Number of Citations
A. Oliveira	52	20
A. Silva	64	38
F. Silva	27	22
J. Oliveira	40	22
J. Silva	35	18
J. Souza	34	12
I. Silva	21	16
M. Silva	21	16
R. Santos	20	17
R. Silva	27	22

3.3 Features

There are different features used by author name disambiguation techniques i.e.,Author Name, Publication Title, Venue Title, Co-authors, Research Area, Email, ORCID, Publication age, Publication Year, Abstract, Full Text, Affiliation etc. Most of the features are extracted from full texts, whereas, some features require external web sources. As discussed in chapter 2, Publication content based and external sources based parameters are not always available. We only used citation meta-data based features for this study, those are most widely used in the literature. These features are:

1. Author Name
2. Venue Title
3. Publication Title
4. Co-Authors

3.4 Preprocessing

Preprocessing step involves the data preparation for training and testing. Dataset files were available in "txt" format. Each text file contains citation records of an ambiguous author group such that each line contains a citation record. Evidences in citation records were separated by semicolons ";".

In preprocessing two steps are involved:

(1) Creation of individual evidences-based datasets and combination-based datasets

(2) Conversion of evidences-based datasets to "arff" format.

First the datasets were divided into different sub-sets for:

(1) individual features i.e., Title, Venue, Co-authors

(2) Combination of the features. i.e., Title-Venue, Title-Coauthors,Venue-Co-authors, Title-Venue-Co-authors

Then the dataset is cleaned and converted to "arff" (Attribute Relation File Format) format accepted by weka. For this purpose, a custom utility was written in python. The utility function for creation of evidences-based datasets and conversion to "arff" format is shown in Code Listing 3.1.

```
def conv_arff(path, dataset, file, ext, names, concat, sep=';',
index_col='None'):
        dir_path='{}/{}'.format(path, dataset)

        df = pd.read_csv('{}/{}.{}'.format(dir_path, file, ext),
        index_col=index_col, sep=sep, names=names, header=None)

        df['data']=df[[col for col in concat]].apply(lambda x:
        ' '.join(x,astype(str)), axis=1)
```

```
df['data']=df['data'].apply(lambda x: '\'{}'\{}'
.format(x.replace(',', ' ')))

p = df[['data', 'class']]
p.to_csv('{}/arff/{}.arff'.format(dir_path, file),
index=None, header=None)

with Prepender('{}/arff/{}.arff'.format(dir_path, file)) as f:
        f.write_lines([
                '@relation \'Dataset {} {}\''.format(dataset, file),
                '@attribute Text string',
                '@attribute class=att {}'.format(set(df['class'])),
                '@data'
        ])
```

LISTING 3.1: Python Utility Function For Evidences-Based Sub-Sets Creation and Conversion to ARFF Format.

3.5 Techniques

This section defines the proposed techniques for the study. For evaluation of evidences we used machine learning classifiers. We transform the AND problem to a single-label multi-class classification task in which a classifier predicts the disambiguated author of a citation. For this purpose, four well-known supervised algorithms (Naïve Bayes, Decision tree, Random Forest, and bagging based ensemble of these classifiers) were used for the classification. The contribution of the evidences is determined by the results of the classification algorithms. The following sub-sections define these classification algorithms.

3.5.1 Naïve Bayes Classification

It is a simple probabilistic algorithm;Naïve Bayes calculates a collection of probabilities by investigating frequency and combination of values in a given data set.

Bayes theorem is applied with the "naïve" assumption that every feature is independent of each other

Suppose C_i be a citation in the ambiguous author group and F be input evidences used in a model assuming all evidences are independent of each other. To predict an author for the citation, a model of Naive Bayes can be defined by

$$P(C_i \mid F) = P(F \mid C_i)xP(C_i)P(N) \tag{3.1}$$

Where $P(C_i \mid F)$ is the posterior probability with variable F that will be C_i.

3.5.2 Decision Tree

The decision tree algorithm is a useful in the classification problem. With this technique, a tree is constructed to model the classification process. It consists of three types of nodes root node, child node, and leaf node. The algorithm starts with defining a root node from the most relationship between every input and output variables. Next, the child node is selected by calculating Information Gain (IG).

$$IG(p,c) = Entropy(p) - [P(x1) * Entropy(x1) + P(x1) * Entropy(x1)] + \cdots \tag{3.2}$$

IG is the information Gain calculated on different possible values or splits, of parent feature (parentNode), for a feature (childNode).

$$Entropy(C_i) = -P(xi)logP(xi)andP(xi) \tag{3.3}$$

Which is the probability of child node i.

Node having the highest IG will become the parent for next generation. This process is repeated until it gets a leaf node and completed decision tree. The

stopping criteria for decision tree is that all the sample for a given node belong to the same class, there aren't remaining attributes for any further partitioning and there aren't any leftover sample. It requires little data preparation.

3.5.3 Random Forest

Random Forest is an ensemble algorithm which was modeled from trees algorithm and Bagging algorithm. It is developed by Breiman. He found that the algorithm can potentially improve classification accuracy. It also works well with a data set with a vast number of input variables. The algorithm begins by creating a combination of trees which each will vote for a class.
Suppose that there are X data and Y input variables in a data set. Let z be the number of sampling groups, xi and yi be a number of data and variables in group i where i is equal to 1, 2, ... and z.
For each xi citation from X. yi variables selected randomly from Y. A tree is grown and gives a prediction class. After Step one to three was recurrent for z times, these trees become a forest. Then the classification will be elected by a majority vote of all trees within the forest.

3.5.4 Bagging - A Voting Based Ensemble Learner

Bagging is one of the most popular classification techniques in ensemble learning. It belongs to type of meta-algorithm in machine learning, specifically designed to improve the generalization and robustness of classification algorithms. In this, several base classifiers of same type are trained independently over distinct bootstrap samples set, the final predicted result is formed by the combination of results for all the base classifiers. For this study, Naive Bayes, Decision Tree and Random Forest algorithms are used as base classifiers. The ensemble prediction is the result of majority voting system which is used as default selection mechanism.

3.6 Evaluation Matrices

Metrics are necessary for empirical analysis of techniques in terms of performance, efficiency and quality. This section defines the matrices used for evaluation and comparison of techniques. We used Precision, Recall, F-Measure and Accuracy. In the following sub sections, these matrices are defined in detail.

3.6.1 Precision

Precision is the fraction of the predicted pairs or clusters in the result that "match" the ground truth for various definitions of what a "match" includes. In other words, precision is a measure of the correctness of the predicted results relative to the ground truth. We can formalize precision based with a widely used statistical approach:

$$P = tp/(tp + fp) \tag{3.4}$$

where tp is short for true positives and fp is short for false positives. True positives are the number of correctly predicted co-references. False positives are the number of falsely predicted co-references.

3.6.2 Recall

Recall (R) is the fraction of truths that are successfully "present"in the result. In other words, recall is a measure of the completeness of the predicted co-references. We can also formalize recall based with a widely used statistical approach:

$$R = tp/(tp + fn) \tag{3.5}$$

3.6.3 F-measure

We note that there are natural trade-offs between precision and recall. For instance, if the result falsely predicts that all the references correspond to the same entity, then the precision would be low as many of these co-references are false, but the recall would be high because all the co-references are captured. Inversely, if the result only predicts a very small number of co-references, then the precision maybe high because the few predicted are correct, but the recall may be low because most co-reference would be missed. To capture this trade-off between precision and recall, the F-Measure, the harmonic mean of the two measures, is often used. F-Measure gives a single measure consisting of features of precision and recall and is computed as follows:

$$F = 2.(P * R)/(P + R) \tag{3.6}$$

3.6.4 Finding Contribution of Evidences

After evaluation of classifiers given above, We will determine the contribution of: (1) each individual evidence i.e, Title, Venue and Co-authors. (2) Combination of evidences i.e., Title-Venue, Venue-Co-authors, Title-Co-authors.

The main idea for calculation of impact of each feature is derived from a game Tug-of-War. In this game two teams pull at opposite ends of a rope until one drags the other over a central line. By each team member has its own impact on the performance, so by adding or removing a member the result changes. We formulated contribution score calculation I_i for each feature i. To determine I_i, we first see the methodology mathematically as:

We have selected three datasets i.e., DBLP, Kisti and BDBComp. So Let

$$D = Setofdatasets \tag{3.7}$$

Whereas

$$\Phi = setofsamples(citations) \tag{3.8}$$

Φ defines the citation records.

$$X = setoffeatures \tag{3.9}$$

X defines the set of features i.e., Title, Venue and Co-authors

$$Y = setoflabels \tag{3.10}$$

Y defines the set of class labels, each unique author is given a class.

$$S =< x, y >: S \in \Phi \tag{3.11}$$

$$\forall_y \in Y \exists_{=1} \tag{3.12}$$

S is defined as a set of feature values from Φ, along with a class label y. And There exist at least one label y.

$$\sigma = Ux_iC^r : 1 < r <= |X| \tag{3.13}$$

Set σ is derived by finding all possible combinations of the features i.e., Title, Venue, CoAuthors, Title-Venue, Title-CoAuthors, Venue-CoAuthors, Title-Venue-CoAuthors. So the set σ, contains subset of citation records according to feature combination with class labels.

$$m = train(f, \sigma))P = test(f^t) \tag{3.14}$$

Training and testing is performed on each of the feature group in σ, 10 Fold cross validation is used for training and testing. F-Measure results are used for for contribution score calculation. Then for each feature group i we derive two set G^i and U^i as:

$$G^i = U_(z \in \sigma) : x_i \in Z and |Z| > 1 \tag{3.15}$$

$$U^i_k = U \forall_e \in G^i_k where e \neq x_i \tag{3.16}$$

Where, G^i is the set of F-measure results from σ, of feature groups containing feature x_i. Whereas, U^i is the set of F-measure results of feature groups from G^i, which do not contain the feature x_i.

Now we can define the Contribution score I_{x_i} as:

$$I(x_i)^n_{i=1} = \sum G^i_k - U^i_k \tag{3.17}$$

We subtract the impact of each feature group i in U^i from G^i, this is partial impact of feature x_i. Add up all the partial impacts to compute the final Contribution score for the feature X_i.

3.7 Summary

This chapter described the Step-by-Step methodology followed for the research. We are using citation metadata-based evidences, available publicly and used by most of the AND techniques i.e., Title, Venue, and Co-authors. We are using DBLP, Kisti, and BDBComp datasets for evaluation of evidences.We transform the AND problem to a single-label multi-class classification task in which a classifier predicts the disambiguated author of a citation. For this purpose, four well-known supervised algorithms (Naïve Bayes, Decision tree, Random Forest, and

bagging based ensemble of these classifiers) were used to do the classification. The contribution of the evidences is determined by the results of the classification algorithms.

Chapter 4

Experiments and Results

4.1 Introduction

The comprehensive methodology adopted for this study is explained in the previous chapter, to find the contribution of evidences in classification for AND. This chapter explains the results achieved by experiments performed using that methodology.

4.2 Preprocessing

We applied the preprocessing steps defined in chapter 3. The steps involved in preprocessing were as follows:

(1) Creation of individual evidences-based datasets and combination-based datasets

(2) Conversion of evidences-based datasets to "arff" format.

A custom utility was written in python for dataset creation and file format conversion. Figure 4.2 Shows the structure of evidences-based dataset folders as a result of preprocessing.

FIGURE 4.1: Evidences-based datasets

4.3 Evaluation

We have selected four well known algorithms for training and testing i.e., Naive Bayes(NB), Decision Tree (J48), Random Forest (RF) and ensemble based bagging of these classifiers (Bagging). Two of them belong to ensemble based algorithms i.e., Random Forest and Bagging, while the rest of the methods are simple classifiers.

Each of these methods is trained and tested over all ambiguous groups of each of the three datasets i.e., DBLP, KISTI, and BDBComp. F-Measure metric was used for the evaluation of the algorithms. Average F-Measure was computed for each individual feature as well as for the groups of features. The results of all algorithms, for each of the three datasets are discussed in the following subsections.

The contribution score was computed from the Average F-Measure results, for each individual feature as well as for the groups of features. The formulas used for the contribution score calculation are discussed in section x.x of chapter 3.

4.4 Algorithms Evaluation

All four algorithms i.e., Naive Bayes, Decision Tree, Random Forest, and Bagging were evaluated over all three datasets i.e., DBLP, Kisti and BDBComp. Average

F-Measure was computed for each individual feature as well as for the groups of features. The results of experiments performed are discussed in detail in the following subsections.

4.4.1 Individual Feature Evaluation

For each dataset i.e., DBLP, Kisti, and BDBComp, All ambigous groups were evaluated for each individual feature i.e., title, venue, and co-authors.

4.4.1.1 DBLP

Table 4.1, shows the Average F-Measure results for DBLP dataset. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48), and Bagging Ensemble (VE) and rows specify the features i.e., Title, Publication Venue and Co-authors. Results shows that the Bagging algorithm has outperformed other algorithms for all three features. For title, Bagging performed 0.36%, 13.17%, 23.69% better than the Naive Bayes, Random Forest and Decision Tree respectively. For publication venue, Bagging performed 0.45%, 6.2%, 14.97% better than the Random Forest, Naive Bayes and Decision Tree respectively. For co-authors, Bagging performed 0.62%, 3.45%, 8.88% better than the Random Forest, Naive Bayes and Decision Tree respectively.

TABLE 4.1: Average F-measure for DBLP Dataset of individual features

Features	**NB**	**J48**	**RF**	**VE**
Title	0.822	0.667	0.729	0.825
Venue	0.629	0.581	0.665	0.668
Co-Authors	0.782	0.743	0.804	0.809

4.4.1.2 Kisti

Table 4.2, shows the Average F-Measure results for Kisti dataset. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision

Tree (J48), and Bagging Ensemble (VE) and rows specify the features i.e., Title, Publication Venue and Co-authors. Overall results of Kisti dataset were less than the results of DBLP dataset, because number of unique authors are more than the number of authors in DBLP dataset hence decreasing the average citations per author. Similar to DBLP, Bagging outperformed other algorithms i.e., Naive Bayes, Random Forest, and Decision Tree. For title, Bagging performed 6.72%, 36.8%, 55.43% better than the Random Forest, Decision Tree and Naive Bayes respectively. For publication venue, Bagging performed 0.64%, 1.51%, 34.29% better than the Naive Bayes, Random Forest and Decision Tree respectively. For co-authors, Bagging performed 1.51%, 3.06%, 22.14% better than the Naive Bayes, Random Forest and Decision Tree respectively.

TABLE 4.2: Average F-Measure Results of Kisti Dataset for individual features

Features	**NB**	**J48**	**RF**	**VE**
Title	0.368	0.418	0.536	0.572
Venue	0.467	0.350	0.463	0.470
Co-Authors	0.663	0.551	0.653	0.673

4.4.1.3 BDBComp

Table 4.3, shows the Average F-Measure results for BDBComp dataset. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48), and Bagging Ensemble (VE) and rows specify the features i.e., Title, Publication Venue and Co-authors. The BDBComp dataset produced lower results than both of the other datasets i.e., DBLP and Kisti, because it is the smallest collection, with many authors having 1 or 2 citation records, so it is very difficult to disambiguate. In contrast with DBLP and Kisti, overall, Naive Bayes performed better than the other algorithms i.e., Bagging, Random Forest, and Decision Tree. For title, Naive Bayes performed 27.68%, 75.71%, 136.54% better than the Bagging, Random Forest and Decision Tree respectively. For publication venue, Bagging performed 2.93%, 17.91%, 26.91% better than the Naive Bayes,

Random Forest and Decision Tree respectively. For co-authors, Naive Bayes performed 13.65%, 15.02%, 74.89% better than the Bagging, Random Forest and Decision Tree respectively.

TABLE 4.3: Average F-Measure of BDBComp Dataset for Individual features

Features	**NB**	**J48**	**RF**	**VE**
Title	0.369	0.156	0.210	0.289
Venue	0.307	0.249	0.268	0.316
Co-Authors	0.383	0.219	0.333	0.337

4.4.2 Group-wise Feature Results

For each dataset i.e., DBLP, Kisti, and BDBComp, All ambiguous groups were evaluated for all possible groups of features i.e., title-venue, title-coAuthors, venue-coAuthors.

4.4.2.1 DBLP

Table 4.4, shows the Average F-Measure results for DBLP dataset. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48), and Bagging Ensemble (VE) and rows specify the feature groups i.e., title-venue, title-coAuthors, venue-coAuthors. Results shows that the Bagging algorithm has outperformed other algorithms for all three feature groups. For title-venue group, Bagging performed 0.36%, 6.59%, 23.69% better than the Naive Bayes, Random Forest and Decision Tree respectively. For title-coAuthors group, Bagging performed 0.34%, 4.05%, 14.87% better than the Naive Bayes, Random Forest and Decision Tree respectively. For venue-coAuthors group, Bagging performed 1.9%, 3.99%, 11.69% better than the Naive Bayes, Random Forest and Decision Tree respectively.

TABLE 4.4: Average F-measure for DBLP Dataset for Groups

Features	**NB**	**J48**	**RF**	**VE**
Title-Venue	0.822	0.667	0.774	0.825
Title-CoAuthors	0.870	0.760	0.839	0.873
Venue-CoAuthors	0.844	0.770	0.827	0.860

4.4.2.2 Kisti

Table 4.5, shows the Average F-Measure results for Kisti dataset. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48), and Bagging Ensemble (VE) and rows specify the feature groups i.e., title-venue, title-coAuthors, venue-coAuthors. Overall, Naive Bayes performed better than other algorithms. For title-venue group, Naive Bayes performed 3%, 13.58%, 50.98% better than the Bagging, Random Forest and Decision Tree respectively. For title-coAuthors group, Naive Bayes performed 0.56%, 7.69%, 31.49% better than the Bagging, Random Forest and Decision Tree respectively. For venue-coAuthors group, Bagging performed 1.43%, 3.95%, 25.84% better than the Naive Bayes, Random Forest and Decision Tree respectively.

TABLE 4.5: Group wise Average F-Measure Results of Kisti Dataset

Features	**NB**	**J48**	**RF**	**VE**
Title-Venue	0.619	0.410	0.545	0.601
Title-CoAuthors	0.714	0.543	0.663	0.710
Venue-CoAuthors	0.701	0.565	0.684	0.711

4.4.2.3 BDBComp

Table 4.6, shows the Average F-Measure results for BDBComp dataset. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48), and Bagging Ensemble (VE) and rows specify the feature groups i.e., title-venue, title-coAuthors, venue-coAuthors. For BDBComp dataset, Naive

Bayes has outperformed other algorithms for all three feature groups. For title-venue group, Naive Bayes performed 23.84%, 56.62%, 75.31% better than the Bagging, Random Forest and Decision Tree respectively. For title-coAuthors group, Naive Bayes and Bagging equally performed 51.7%, 71.54% better than the Random Forest and Decision Tree respectively. For venue-coAuthors group, Naive Bayes performed 9.85%, 26.09%, 67.31% better than the Bagging, Random Forest and Decision Tree respectively.

TABLE 4.6: Average F-Measure of BDBComp Dataset for Groups

Features	**NB**	**J48**	**RF**	**VE**
Title-Venue	0.426	0.243	0.272	0.344
Title-CoAuthors	0.446	0.260	0.294	0.446
Venue-CoAuthors	0.435	0.260	0.345	0.396

In Table 4.7, The average F-Measure results are shown for the experiments performed using group of all three features i.e., Title, Publication Venue and Co-authors. The columns specify the algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48), and Bagging Ensemble (VE) and rows specify the datasets used for the experiment i.e, DBLP, Kisti, and BDBComp. Overall, The results of DBLP dataset were better the kisti and BDBComp datasets for all algorithms except Naive Bayes. Bagging performed 0.11%, 5.4%, 77.82% better than the Random Forest res, J48 and Naive bayes in DBLP dataset respectively. Naive Bayes performed 0.56%, 7.69%, 31.49% better than the bagging, decision tree, and random forest in Kisti dataset respectively. In BDBComp dataset, Naive bayes performed 10.59%, 40.33%, 64.62% better than the bagging, decision tree, and random forest respectively.

TABLE 4.7: Average F-Measure for Title-Venue-CoAuthors for all datasets

Dataset	NB	RF	J48	VE
DBLP	0.5052	0.8974	0.8517	0.8981
Kisti	0.7138	0.5433	0.6625	0.7101
BDBComp	0.4283	0.2596	0.3046	0.3871

4.5 Contribution Score Evaluation

From the results, it can be seen that the performance of algorithms is significantly changed with the usage of different feature groups. Different feature groups have their own impact in the overall results. The following subsections discuss the contribution score of each individual feature and each feature group. The contribution scores are computed from the Average F-Measure results discussed in section 4.5.

4.6 Individual Feature Contribution Score

Contribution score was computed for all individual features i.e., title, publication venue and co-authors, for each dataset i.e., DBLP, Kisti and BDBComp.

4.6.0.1 DBLP

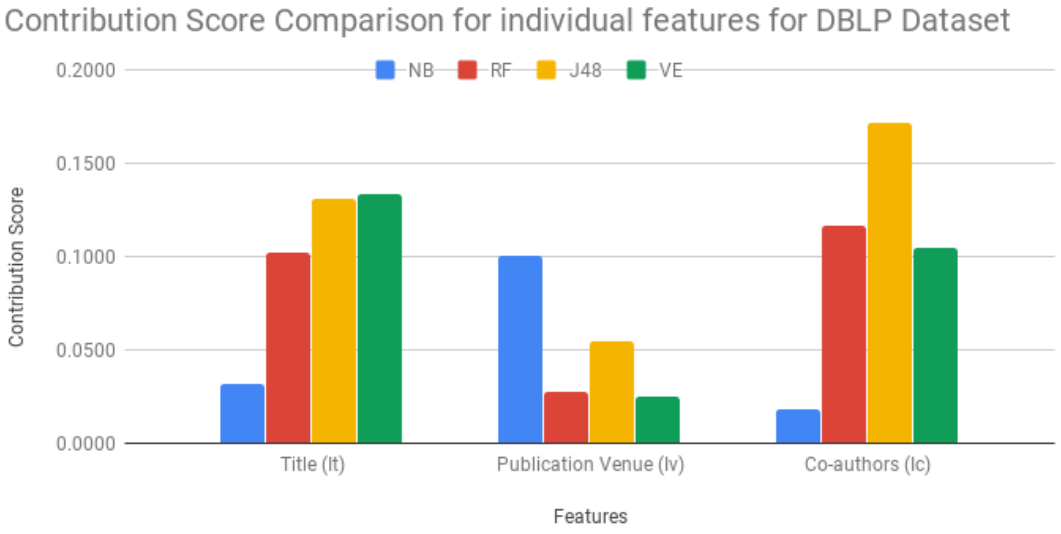

FIGURE 4.2: Contribution Score of individual features for DBLP Dataset

Figure 4.2, shows the comparison of contribution score of individual features in DBLP dataset for all four algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48) and Bagging (VE) represented with blue, red, yellow and green

colour bars respectively. X-Axis represents the features and Y-Axis represents the contribution score.

Different algorithms show different trends of the contribution of individual features for DBLP dataset. Co-authors outperformed title and publication venue in tree based classifiers. In DBLP dataset, number of citation records available per author are 18 whereas, Kisti and BDBComp have 15, 1.67 citations per author respectively. Co-authors and title features have diverse record, whereas, publication venue feature have less diverse records. Tree based algorithms such as Decision tree and and random forest algorithms performs well on diverse features. Co-authors feature contributed 13.88%,57.17% more than title and publication venue in random forest and decision tree algorithms respectively. It contributed 78.77%, 27.36% less than title in Naive Bayes and Bagging respectively. In Naive Bayes, Publication Venue outperformed title and co-authors by 214.69%, 462.57% respectively. Naive Bayes is a probabilistic classifier, it works well with the less diverse features.

4.6.0.2 Kisti

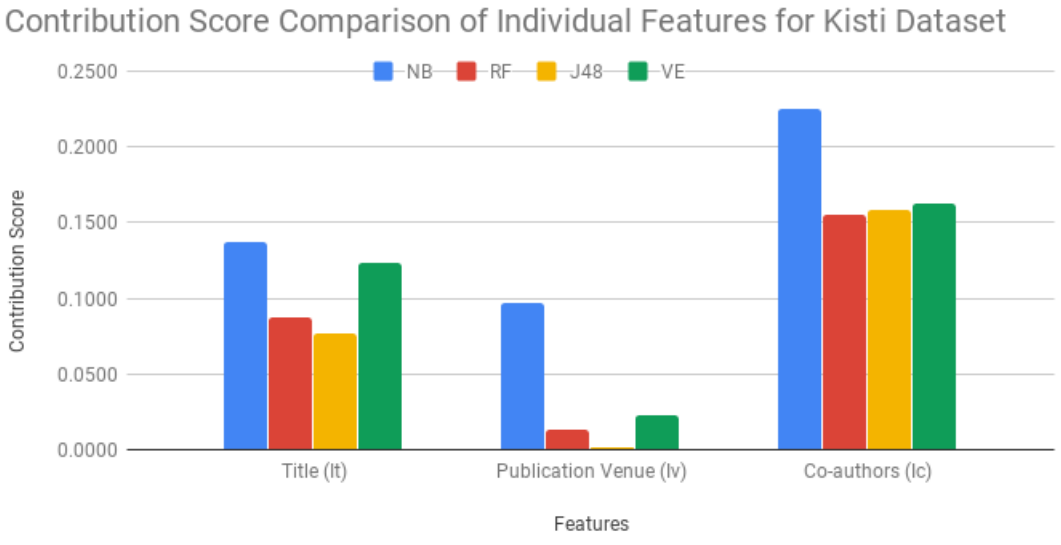

FIGURE 4.3: Contribution Score of individual features for Kisti Dataset

Figure 4.3, shows the comparison of contribution score of individual features in Kisti dataset for all four algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48) and Bagging (VE) represented with blue, red, yellow and green colour bars respectively. X-Axis represents the features and Y-Axis represents the contribution score.

In Kisti dataset, overall, Co-authors has contributed more than title and publication venue in all algorithms. In Kisti dataset, Co-authors and title features have less diverse records then the DBLP dataset, due to which Naive bayes outperformed other methods for all three features. Co-authors feature contributed 64.09%, 31.98%, 78.57%, 104.4% more than title in Naive Bayes, Bagging, random forest and decision tree algorithms respectively. Co-authors feature also contributed 133%, 624.4%, 1056.72%, 7790% more than publication venue in Naive Bayes, Bagging, random forest and decision tree algorithms respectively. Contribution of title in Naive Bayes is 11.17%, 58.18%, 77.85% more than bagging, random forest and decision tree. Naive Bayes also performed well for publication venue, publication venue contributed 329.7%, 621%, 4735% more than the bagging, rendom forest and decision tree.

4.6.0.3 BDBComp

Figure 4.4, shows the comparison of contribution score of individual features in BDBComp dataset for all four algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48) and Bagging (VE) represented with blue, red, yellow and green colour bars respectively. X-Axis represents the features and Y-Axis represents the contribution score.

BDBComp dataset is the smallest dataset, overall, Co-authors has contributed more than title and publication venue in all algorithms. Co-authors feature contributed 88%,1691%,2793% more than title in Bagging, random forest and decision tree algorithms respectively. Co-authors feature also contributed 123.62%, 407%, 127%, 2% more than publication venue in Naive Bayes, Bagging, random forest and decision tree algorithms respectively. Contribution of title in Naive Bayes is

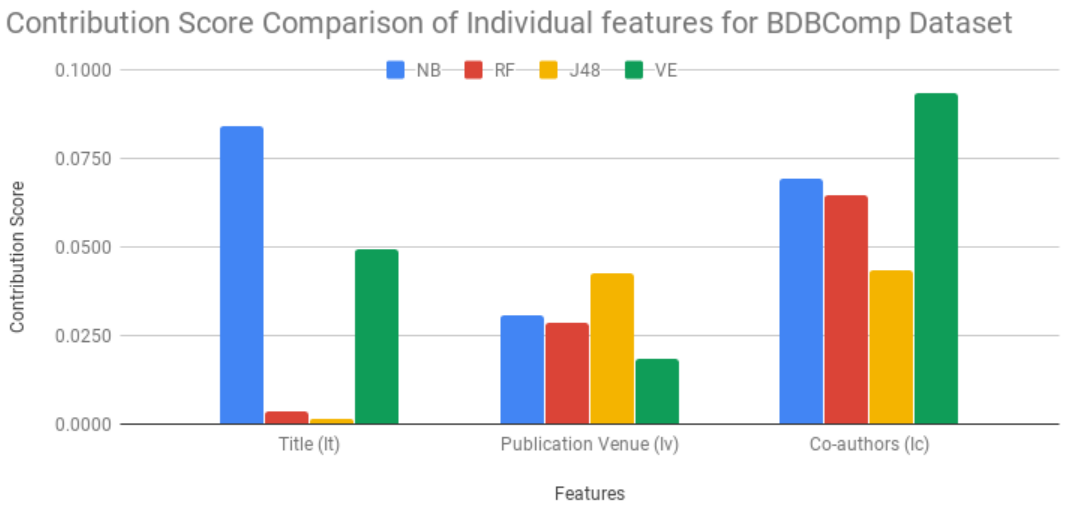

FIGURE 4.4: Contribution Score of individual features for BDBComp Dataset

70%, 2236%, 5506% more than bagging, random forest and decision tree. Publication venue contributed 37.54%, 49%, 130% more in decision tree than the Naive Bayes, random forest and bagging.

4.7 Feature Groups Contribution Score

Contribution score was computed for all groups of features i.e., title-venue, title-coAuthors, and venue-coAuthors, for each dataset i.e., DBLP, Kisti and BDB-Comp.

4.7.0.1 DBLP

Figure 4.5, shows the comparison of contribution score of group of features in DBLP dataset for all four algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48) and Bagging (VE) represented with blue, red, yellow and green colour bars respectively. X-Axis represents the feature groups and Y-Axis represents the contribution score.

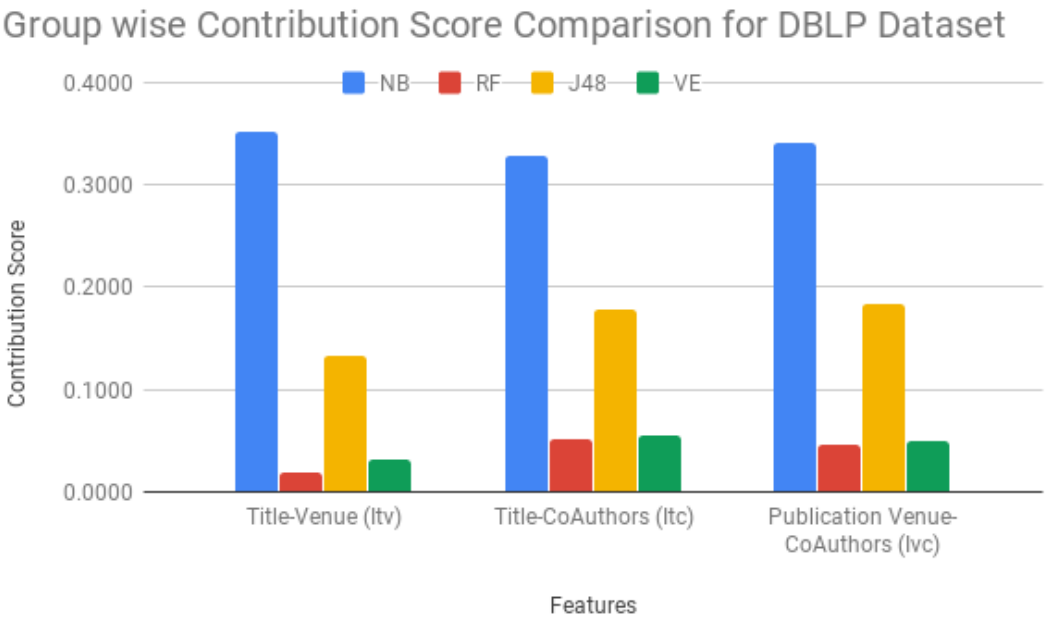

FIGURE 4.5: Contribution Score of individual features for DBLP Dataset

Naive Bayes outperformed all other methods for all feature groups. For Naive Bayes, title-venue group contributed 166%, 1020%, 1781% more than decision tree, bagging and random forest. title-coAuthors group contributed 83%, 488%, 544% more than decision tree, bagging and random fores. Venue-coAuthors group contributed 85% ,592%, 652% more than decision tree, bagging and random fores.

4.7.0.2 Kisti

Figure 4.6, shows the comparison of contribution score of group of features in Kisti dataset for all four algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48) and Bagging (VE) represented with blue, red, yellow and green colour bars respectively. X-Axis represents the feature groups and Y-Axis represents the contribution score.

In kisti dataset, contribution of groups containing co-authors i.e., title-coAuthors and venue-coAuthors is more than than the title-venue group for all methods. For Naive bayes, title-coAuthors contributed 13.05% and 766 % more than venue-coAuthors and title-venue. For Random Forest, venue-coAuthors contributed 22%

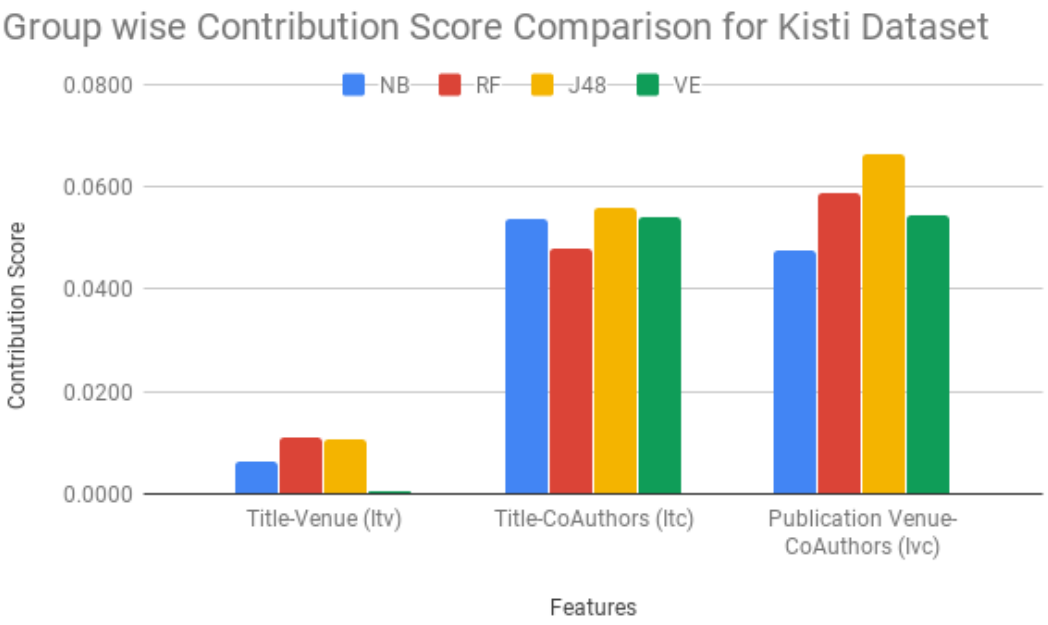

FIGURE 4.6: Contribution Score of individual features for Kisti Dataset

and 483 % more than the title-coAuthors and title-venue groups. For decision tree, venue-coAuthors contributed 19% and 516% more than the title-coAuthors and title-venue groups. For bagging, venue-coAuthors and title-coAuthors equally contributed 8900 % more than title-venue group.

4.7.0.3 BDBComp

Figure 4.7, shows the comparison of contribution score of group of features in BDBComp dataset for all four algorithms i.e., Naive Bayes (NB), Random Forest (RF), Decision Tree (J48) and Bagging (VE) represented with blue, red, yellow and green colour bars respectively. X-Axis represents the feature groups and Y-Axis represents the contribution score.

BDBComp shows different feature group contribution trends, overall, title-venue contributed more than title-coAuthors and venue-coAuthors group. For Bagging, title-venue contributed 95%, 346% more than the title-coAuthors and venue-coAuthors. For random forest, venue-coAuthors contributed 51%, 528% more than

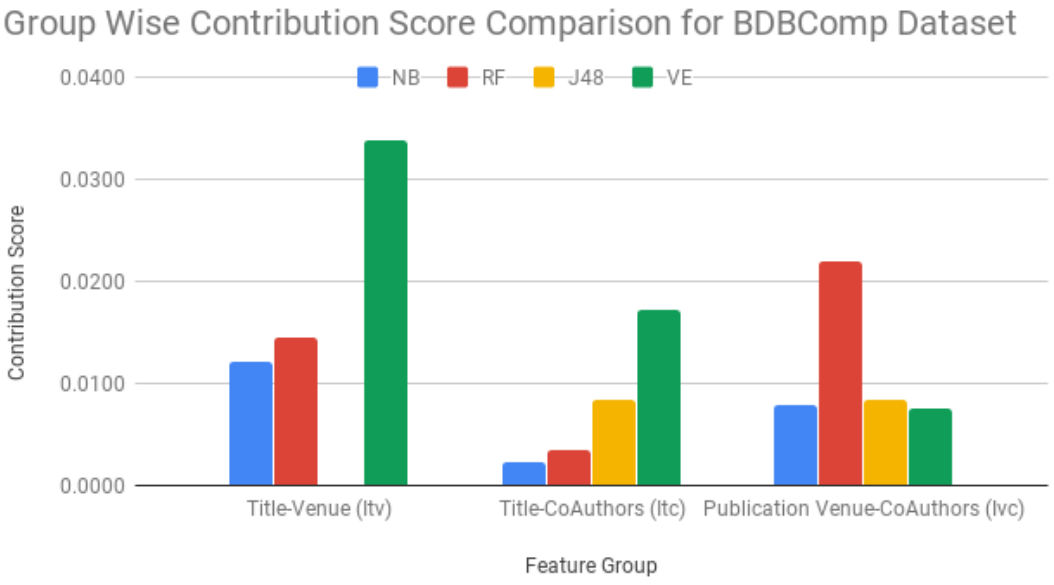

FIGURE 4.7: Contribution Score of individual features for BDBComp Dataset

the title-venue and title-coAuthors groups. Title-venue group contributed 55.13%, 426% more than title-coAuthors and venue-coAuthors for Naive Bayes classifier.

Results show different feature contribution trends for different algorithms. Overall, For the smaller sets, such as BDBComp and Kisti, it is better to use Naive Bayes, when you have title and co-author features available, because Naive Bayes performs better for pattern identification from text based features and it also performs good for co-authorship pattern when there are significant co-authors available. For larger and diverse datasets like DBLP, Ensemble based algorithms i.e, Random Forest and Bagging provide better results because it performs voting among multiple possible patterns and classifiers. Ensemble based algorithms also reduce biasness caused by training set. In case of small datasets i.e., BDBComp, nominal attributes like publication venue and co-authors , decision tree or decision tree based ensemble i.e., Random Forest performs better.

4.8 Summary

Results show different feature contribution trends for different algorithms. Overall, For the smaller sets, such as BDBComp and Kisti, it is better to use Naive Bayes, specially when you have title and co-author features available. For larger and diverse datasets like DBLP, tree or ensemble based algorithms provide better results.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Authors contribute to the scientific society by publishing their work in online scientific data management systems known as digital libraries. Because, such system are not following a unique schema, there is no unique identifier for authors and publications. Publications are indexed by author names. Author name ambiguities arise due to natural limitation of names, ambiguities like polysem and synonym occurs. The polysem refers to the same author publishing using different name variants. Whereas synonym refers to the phenomena when different authors publish using the same name. The problem of assigning true authors to their own citations and publications is known as author name disambiguation.

Many different methods found in the literature categorized as Machine learning based methods including supervised, unsupervised, and semi-supervised methods , heuristic based methods and graph based methods. Various methods have utilized the different set of features for author name disambiguation such as title, Co-authors, Publication Venues, affiliations, keywords, abstracts, publication years, publication age, topic models etc.. These features are usually not available at once.

In this study, three most widely used features in literature were used i.e., title, publication venue, and co-author names. A citation record contains all these features. Four popular machine learning methods were selected for the study i.e., Naive Bayes, Random Forest, Decision Tree and Bagging. From the literature, Three most popular datasets were chosen for evaluation.

Each dataset was composed of different ambiguous groups. Different subsets of each dataset were generated for individual features evaluation and feature group evaluation. For this purpose a custom utility was written in python. All four methods were evaluated using a popular machine learning tool WEKA, for each subset to compute the F-Measure results. Ten Fold cross-validation was applied for training and testing of methods. Average F-Measure was computed for each dataset. We derived contribution score calculation formulas. Average F-Measure is used for contribution score calculation.

The source code and datasets is made publically available at GitHub, to help the interested research community in this domain.

The findings of this study are as under:

1. The study comprehensively discussed the contribution score of each individual feature as well as for the feature groups. Which is also the answer to our research question 1 (RQ1).

2. Four selected methods were comprehensively evaluated for individual features as well as for the feature groups. The cases are discussed in detail in results section. Which is also the answer to our research question 2 (RQ2).

3. Results show different feature contribution trends for different algorithms. Overall, For the smaller sets, such as BDBComp and Kisti, it is better to use Naive Bayes, specially when you have title and co-author features available. For larger and diverse datasets like DBLP, tree or ensemble based algorithms provide better results.

5.2 Future Work

This study is focused on the most widely used and freely available feature set in the author name disambiguation domain. The research can be extended to find contribution of the other explicit features such as keywords, abstracts, affiliations, publication years etc., as well as of the implicit features such as publication age, topic models etc.

The methodology can be extended to find the impact of features in other domains, such as biology, chemistry, Physics, Bio-Informatics etc.

Bibliography

[1] M. Farooq, H. U. Khan, S. Iqbal, E. U. Munir, and A. Shahzad, "Ds-index: Ranking authors distinctively in an academic network," *IEEE Access*, vol. 5, pp. 19 588–19 596, 2017.

[2] I. Hussain and S. Asghar, "A survey of author name disambiguation techniques: 2010–2016," *The Knowledge Engineering Review*, vol. 32, 2017.

[3] P. Mitra, J. Kang, D. Lee, and B.-w. On, "Comparative study of name disambiguation problem using a scalable blocking-based framework," in *Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference.* IEEE, 2005, pp. 344–353.

[4] S. Elliot, "Survey of author name disambiguation: 2004 to 2010," 2010.

[5] L. V. B. Esperidião, A. A. Ferreira, A. H. Laender, M. A. Gonçalves, D. M. Gomes, A. I. Tavares, and G. T. de Assis, "Reducing fragmentation in incremental author name disambiguation," *Journal of Information and Data Management*, vol. 5, no. 3, pp. 293–293, 2014.

[6] R. Hazra, A. Saha, S. B. Deb, and D. Mitra, "An efficient technique for author name disambiguation," in *Current Trends in Advanced Computing (ICCTAC), IEEE International Conference.* IEEE, 2016, pp. 1–6.

[7] F. Momeni and P. Mayr, "Using co-authorship networks for author name disambiguation," in *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries.* ACM, 2016, pp. 261–262.

[8] M.-C. Müller, F. Reitz, and N. Roy, "Data sets for author name disambiguation: an empirical analysis and a new resource," *Scientometrics*, vol. 111, no. 3, pp. 1467–1500, 2017.

[9] E. Bastrakova, R. Ledesma, J. Millan, F. Rico, and D. Zighed, "Relational machine learning author disambiguation," in *Artificial Intelligence and Natural Language Conference (AINL), IEEE*. IEEE, 2016, pp. 1–7.

[10] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "A brief survey of automatic methods for author name disambiguation," *Acm Sigmod Record*, vol. 41, no. 2, pp. 15–26, 2012.

[11] 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.

[12] H. J. Si, W. Tong, and S. Kausar, "A conditional random field model for name disambiguation in national natural science foundation of china fund," *Journal of Algorithms & Computational Technology*, vol. 12, no. 2, pp. 91–100, 2018.

[13] A. Daud, N. R. Aljohani, R. A. Abbasi, Z. Rafique, T. Amjad, H. Dawood, and K. H. Alyoubi, "Finding rising stars in co-author networks via weighted mutual influence," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 33–41.

[14] T. Gurney, E. Horlings, and P. Van Den Besselaar, "Author disambiguation using multi-aspect similarity indicators," *Scientometrics*, vol. 91, no. 2, pp. 435–449, 2012.

[15] D. Han, S. Liu, Y. Hu, B. Wang, and Y. Sun, "Elm-based name disambiguation in bibliography," *World Wide Web*, vol. 18, no. 2, pp. 253–263, 2015.

[16] 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.

[17] H. N. Tran, T. Huynh, and T. Do, "Author name disambiguation by using deep neural network," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2014, pp. 123–132.

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

[19] H. Han, W. Xu, H. Zha, and C. L. Giles, "A hierarchical naive bayes mixture model for name disambiguation in author citations," in *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 2005, pp. 1065–1069.

[20] H. Wu, B. Li, Y. Pei, and J. He, "Unsupervised author disambiguation using dempster–shafer theory," *Scientometrics*, vol. 101, no. 3, pp. 1955–1972, 2014.

[21] J. Tang, A. C. 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.

[22] D. Shin, T. Kim, J. Choi, and J. Kim, "Author name disambiguation using a graph model with node splitting and merging based on bibliographic information," *Scientometrics*, vol. 100, no. 1, pp. 15–50, 2014.

[23] B.-W. On, E. Elmacioglu, D. Lee, J. Kang, and J. Pei, "Improving grouped-entity resolution using quasi-cliques," in *ICDM'06. Sixth International Conference on Data Mining*. IEEE, 2006, pp. 1008–1015.

[24] H.-T. Peng, C.-Y. Lu, W. Hsu, and J.-M. Ho, "Disambiguating authors in citations on the web and authorship correlations," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10 521–10 532, 2012.

[25] X. Wang, J. Tang, H. Cheng, and S. Y. Philip, "Adana: Active name disambiguation," in *IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 794–803.

[26] 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. 332–332, 2012.

[27] D. A. Pereira, B. Ribeiro-Neto, N. Ziviani, A. H. Laender, M. A. Gonçalves, and A. A. Ferreira, "Using web information for author name disambiguation," in *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2009, pp. 49–58.

[28] Q. Shen, T. Wu, H. Yang, Y. Wu, H. Qu, and W. Cui, "Nameclarifier: A visual analytics system for author name disambiguation," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 141–150, 2017.

[29] D. Lee, B.-W. On, J. Kang, and S. Park, "Effective and scalable solutions for mixed and split citation problems in digital libraries," in *Proceedings of the 2nd international workshop on Information quality in information systems*. ACM, 2005, pp. 69–76.

[30] N. Onodera, M. Iwasawa, N. Midorikawa, F. Yoshikane, K. Amano, Y. Ootani, T. Kodama, Y. Kiyama, H. Tsunoda, and S. Yamazaki, "A method for eliminating articles by homonymous authors from the large number of articles retrieved by author search," *Journal of the Association for Information Science and Technology*, vol. 62, no. 4, pp. 677–690, 2011.

[31] T. Huynh, K. Hoang, T. Do, and D. Huynh, "Vietnamese author name disambiguation for integrating publications from heterogeneous sources," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2013, pp. 226–235.

[32] A. P. de 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. 289–289, 2011.

[33] L. Tang and J. P. Walsh, "Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps," *Scientometrics*, vol. 84, no. 3, pp. 763–784, 2010.

[34] M. Imran, S. Z. H. Gillani, M. Marchese *et al.*, "A real-time heuristic-based unsupervised method for name disambiguation in digital libraries," *D-Lib Magazine*, vol. 19, no. 9, pp. 1–1, 2013.

[35] C. Schulz, A. Mazloumian, A. M. Petersen, O. Penner, and D. Helbing, "Exploiting citation networks for large-scale author name disambiguation," *EPJ Data Science*, vol. 3, no. 1, pp. 11–11, 2014.

[36] Y. Liu, W. Li, Z. Huang, and Q. Fang, "A fast method based on multiple clustering for name disambiguation in bibliographic citations," *Journal of the Association for Information Science and Technology*, vol. 66, no. 3, pp. 634–644, 2015.

[37] J. Zhu, X. Wu, X. Lin, C. Huang, G. P. C. Fung, and Y. Tang, "A novel multiple layers name disambiguation framework for digital libraries using dynamic clustering," *Scientometrics*, vol. 114, no. 3, pp. 781–794, 2018.

[38] J. Zhao, P. Wang, and K. Huang, "A semi-supervised approach for author disambiguation in kdd cup 2013," in *Proceedings of the 2013 KDD CUP 2013 Workshop*. ACM, 2013, pp. 10–10.

[39] G. Louppe, H. T. Al-Natsheh, M. Susik, and E. J. Maguire, "Ethnicity sensitive author disambiguation using semi-supervised learning," in *International Conference on Knowledge Engineering and the Semantic Web*. Springer, 2016, pp. 272–287.

[40] Y. Zhu and Q. Li, "Enhancing object distinction utilizing probabilistic topic model," in *International Conference on Cloud Computing and Big Data (CloudCom-Asia)*. IEEE, 2013, pp. 177–182.

[41] P. Wang, J. Zhao, K. Huang, and B. Xu, "A unified semi-supervised framework for author disambiguation in academic social network," in *International*

Conference on Database and Expert Systems Applications. Springer, 2014, pp. 1–16.

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

[43] W.-S. Chin, Y. Zhuang, Y.-C. Juan, F. Wu, H.-Y. Tung, T. Yu, J.-P. Wang, C.-X. Chang, C.-P. Yang, W.-C. Chang *et al.*, "Effective string processing and matching for author disambiguation," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3037–3064, 2014.

[44] A. F. Santana, M. A. Gonçalves, A. H. Laender, and A. A. Ferreira, "Incremental author name disambiguation by exploiting domain-specific heuristics," *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 931–945, 2017.

[45] D. Vishnyakova, R. Rodriguez-Esteban, K. Ozol, and F. Rinaldi, "Author name disambiguation in medline based on journal descriptors and semantic types," in *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, 2016, pp. 134–142.

[46] J. Zhu, Y. Yang, Q. Xie, L. Wang, and S.-U. Hassan, "Robust hybrid name disambiguation framework for large databases," *Scientometrics*, vol. 98, no. 3, pp. 2255–2274, 2014.

[47] J. Schulz, "Using monte carlo simulations to assess the impact of author name disambiguation quality on different bibliometric analyses," *Scientometrics*, vol. 107, no. 3, pp. 1283–1298, 2016.

[48] A. A. Ferreira, M. A. Gonçalves, J. M. Almeida, A. H. Laender, and A. Veloso, "Sygar–a synthetic data generator for evaluating name disambiguation methods," in *International Conference on Theory and Practice of Digital Libraries*. Springer, 2009, pp. 437–441.

[49] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "Automatic methods for disambiguating author names in bibliographic data repositories," in *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, 2015, pp. 297–298.

[50] I.-S. Kang, P. Kim, S. Lee, H. Jung, and B.-J. You, "Construction of a large-scale test set for author disambiguation," *Information Processing & Management*, vol. 47, no. 3, pp. 452–465, 2011.