INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

An Efficient Probabilistic Framework For Name Disambiguation In Digital Library

D.Lourdu Sophia
Final Year M.E(Computer Science And Engineering)
Arunai Engineering College

Abstract:

Despite years of research, the name ambiguity problem remains largely unresolved. Outstanding issues include how to capture all information for name disambiguation in a unified approach, and how to determine the number of people K in the disambiguation process. In this paper, we formalize the problem in a unified probabilistic framework, which incorporates both attributes and relationships. Specifically, we define a disambiguation objective function for the problem and propose a two-step parameter estimation algorithm. We also investigate a dynamic approach for estimating the number of people K. Experiments show that our proposed framework significantly outperforms four baseline methods of using clustering algorithms and two other previous methods. Experiments also indicate that the number K automatically found by our method is close to the actual number.

Key words*: Digital libraries, information search and retrieval, database applications, heterogeneous databases.*

1.Introduction

DIFFERENT people may share identical names in the real world. It is estimated that the 300 most common male names are used by more than 114 million people (taking about 78.74 percent) in the United States (http://names. mongabay.com/male_names.htm). In many applications such as scientific literature management and information integration, the people names are used as the identifier to retrieve the information. Name ambiguity will greatly hurt the quality of the retrieved information. To underline the seriousness of the problem, we have examined 100 person names in the publication data and found, for example, there are 54 papers authored by 25 different "Jing Zhang" in the DBLP database. Also, three students named "Yi Li" have graduated from the first author's lab.

1.1.Motivation

We begin by illustrating the problem with an example drawn from a real-world system (http://arnetminer.org) [40]. In this system, we try to extract researcher profiles from the web and integrate the publication data from online databases such as DBLP, ACM Digital Library, Cite Seer, and SCI. In the integration, we inevitably have the name ambiguity problem. Fig. 1 shows a simplified example. In Fig. 1, each node denotes a paper (with title omitted). Each directed edge denotes a relationship between two papers with a label representing the type of the relationship (cf. Section 2.1 for definitions of the relationship types). The distance between two nodes denotes the similarity of the two papers in terms of some content-based similarity measurement (e.g., cosine similarity). The solid polygon outlines the ideal disambiguation results, which indicate that 11 papers should be assigned to three different authors. An immediate observation from Fig. 1 is that a method based on only content similarity (the distance) would be difficult to achieve satisfactory performance, and that different types of relationships can be helpful, but with different degrees of contribution. For example, there is a CoAuthor relationship between nodes #3 and #8. Although the similarity between the two nodes is not high, benefiting from the CoAuthor relationship, we can still assign the two nodes (papers) to the same author. On the contrary, although there is a Citation relationship between nodes #3 and #7, the two papers are assigned to two different authors. Thus, one challenge here is how to design an algorithm for the name disambiguation problem by considering both attribute information of the node and the relationships between nodes.

1.2.Prior Work

The problem has been independently investigated in different domains, and is also known as entity resolution [4], [5], [7], web appearance disambiguation [3], [20], name identification [26], and Object distinction [49]. Despite many approaches proposed, the name ambiguity problem remains largely unresolved. In general, existing methods for name disambiguation mainly fall into three categories: supervised based, unsupervised based, and constraint based. The supervised-based approach (e.g., [17]) tries to learn a specific classification model for each author name from the human labeled training data. Then, the learned model is used to predict the author assignment of each paper. In the unsupervised based approach (e.g., [18], [36], [37], [49]), clustering algorithms or topic models are employed to find paper partitions, and papers in different partitions are assigned to different authors. The constraint-based approach also

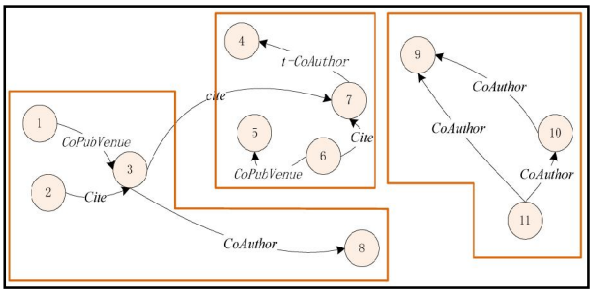

Figure 1

utilizes the clustering algorithms. The difference is that user provided constraints are used to guide the clustering algorithm toward better data partitioning (e.g., [2], [51]).

Furthermore, several other approaches based on rules, citation/author graphs, and combinations of the different approaches have been studied. For example, Whang et al. [47] introduce a negative rules-based approach to remove the inconsistencies in the databases and develop two algorithms to identify important properties to create the rules. Davis et al. [11] have developed an interactive system which permits a user to locate the occurrences of named entities within a given text. The system is to identify references to a single art object (e.g., a particular building) in text related to images of that object in a digital collection. McRae-Spencer and Shadbolt [28] present a graph-based approach to author disambiguation on large-scale citation networks by using self-citation, coauthor relationships. The approach can achieve a high precision but a relatively low recall. Yu et al. [50] have developed supervised approaches to identify the full forms of ambiguous

abbreviations within the context they appear. More recently, Chen et al. [8] study how to combine the different disambiguation approaches and propose an entity resolution ensemble framework, which combines the results of multiple base-level entity resolution systems into a single solution to improve the accuracy of entity resolution. Whang et al. [46] propose an iterative blocking framework where the resolution results of blocks are reflected to subsequently processed blocks. On and Lee [32] study the scalability issue of the name disambiguation problem. Although much progress has been made, existing methods do not achieve satisfactory disambiguation results due to their limitations:

- Some existing graph clustering methods (e.g., [31], [35], [48]) focus on partitioning the data graph based on the topological structure; some other methods (e.g., [18], [42]) aim to cluster the data graph according to node similarity. A few researchers (e.g., [38], [52]) try to combine the two pieces of information. For example, Zhou et al. attempt to combine information based on both vertex attributes (i.e., node similarity) and graph topological structure by first constructing an attribute augmented graph through explicit assignments of attribute, value pairs to vertices, and subsequently estimating the pair wise vertices' closeness using a random walk model. The pair wise comparisons mean that they subsequently discard topological information. Although the authors were able to demonstrate that attribute similarity increases the closeness of pair wise vertices in their distance measure, how to optimally balance the contributions of the different information is still an open problem They are only able to conclude that adding attribute similarity information to the clustering objective will not degrade the intra cluster closeness. Further, in [52], the experimental data sets contain very few attributes. The first data set (political blogs) only has one (binary) attribute and the second data set of DBLP bibliographical data only has two attributes. We argue that much richer node attribute information is required for tackling the name disambiguation problem effectively.
- The performance of all the aforementioned methods depends on accurately estimating K. Although several clustering algorithm such as X-means [33] can automatically find the number K based on some splitting criterion, it is unclear whether such a method can be directly applied to the name disambiguation problem.

- In exiting methods, the data usually only contain homogeneous nodes and relationships; while in our problem setting, there may be multiple different relationships (e.g., CoAuthor and Citation) between nodes. The types of different relationships may have different importance for the name disambiguation problem. How to automatically model the degree of contributions of different relationships is still a challenging problem.

1.3.Our Solution

Having conducted a thorough investigation, we propose a unified probabilistic framework to address the above challenges. Specifically, we formalize the disambiguation problem using a Markov Random Fields (MRF) [16], [24], in which the data are cohesive on both local attributes and relationships. We explore a dynamic approach for estimating the number of people K and a two-step algorithm for parameter estimation. The proposed approach can achieve better performance in name disambiguation than existing methods because the approach takes advantage of interdependencies between paper assignments. To the best of our knowledge, our work is the first to formalize all the problems for name disambiguation in a unified framework and tackle the problems together.

The proposed framework is quite general. One can incorporate any relational features or local features into the framework, e.g., a feature based on the web search engine used. The framework can be also extended to deal with many other problems such as entity resolution in a relational database [4].

Our contributions in this paper include: 1) formalization of the name disambiguation problem in a unified probabilistic framework; 2) proposal of an algorithm to solve the parameter estimation in the framework; and 3) an empirical verification of the effectiveness of the proposed framework.

2.Problem Formalization

2.1.Definitions

- In the discussion that follows, we assign six attributes to each paper pi as shown in Table 1. Such publication data can be extracted from sources such as DBLP, Libra.msra.cn, Arnetminer.org, and Citeseer.ist.psu.edu.

Attribute	Description
$p_i.title$	title of p_i
$p_i.pubvenue$	published conference/journal of p_i
$p_i.year$	published year of p_i
$p_i.abstract$	abstract of p_i
$p_i.authors$	authors name set of p_i $\{a_i^{(0)},a_i^{(1)},\ldots a_i^{(a)}\}$
$p_i.references$	references of p_i

Table 1: Attributes Of Each Publication Pi

2.1.1.Definition 1 (Principle Author And Secondary Author)

Each paper pi has one or more authors We describe the author name that we are going to disambiguate as the principle author and the rest (if any) as secondary authors.

We define five types of undirected relationships between papers (Table 2). Specifically, Co Pub Venue ðr1Þ represents two papers published at the same venue. For example, if both papers are published at "KDD," we create an undirected Co Pub Venue relationship between the two papers. Intuitively, two researchers with the same name may work in different research fields, thus would publish papers at different venues.

- CoAuthor ðr2Þ represents that two papers p1 and p2 have a secondary author with the same name typically, two papers that have many common coauthors would belong to the same person.
- Citation (r3) represents one paper citing another paper. It is likely that an author cites his own previous work. Further, we incorporate latent citation information as follows: If paper p1 cites papers p2; p3; . . . ; pn, then we establish undirected pair wise relationships among all cited papers, in addition to directed pair wise relationships between p1 and the cited papers.
- Constraint (r4) denotes constraints supplied via user feedback. For instance, the user can specify that two papers should be disambiguated to the same person or should belong to different persons.
- CoAuthor (r5) represents _-extension CoAuthor relationship. We use an example to explain this relationship. Suppose paper pi has authors "David Mitchell" and "Andrew Mark," and has authors "David Mitchell" and "Fernando Mulford." We are going to disambiguate "David Mitchell." And if "Andrew Mark" and "Fernando Mulford" also coauthor another paper, then we say pi and pj have a CoAuthorrelationship.

R	*W*	Relation Name	Description
r_1	w_1	CoPubVenue	$p_i.pubvenue = p_j.pubvenue$
r_2	w_2	CoAuthor	$\exists r, s>0, a_i^{(r)}=a_j^{(s)}$
r_3	w_3	Citation	p_i cites p_j or p_j cites p_i
r_4	w_4	Constraint	feedback supplied by users
r_5	w_5	τ-CoAuthor	τ-extension co-authorship (τ>1)

Table 2: Relationships Between Papers

To make it clear, we explain further about how to determine whether two papers have a _-CoAuthor relationship. From the entire paper data set, we can construct a coauthor network, where each node denotes an author name and each edge denotes a coauthor relationship. For any two papers p1 and p2, we can obtain their corresponding sets A0 p1 and A0 p2 by their coauthors. If and only if we say the two papers have a CoAuthor relationship. For determining a 2-extension CoAuthor relationship, we construct two coauthor sets A2p 1 and A2p 2 according to the coauthor network.Specifically, A2p 1 is the set of authors by extending A0 p1 with all neighbors of the authors in A0 p1, i.e., A2p 1 where NB(a) is the set of neighbors of node a. Then, we say the two papers p1 and p2 have a 2 CoAuthor relationship, if and only if A2p 1 \ A2p 2 6¼ _. For determining whether two papers have a 3-extension CoAuthor relationship, we further extend A2p 1 to find an author set A3p for each paper and if the two sets have an intersection, we say the two papers have a 3- CoAuthor relationship. The weight of each type of relationship ri is denoted by wi. Estimation of the value of different weights will be described in Section 4.

In the name disambiguation problem, some papers may easily be clustered together or may be assigned together by the user. These papers will not be partitioned in the disambiguation algorithm. We describe such group of papers as cluster atom.

2.1.2.Definition 2 (Cluster Atom)

A cluster atom is a cluster in which papers are closely connected (e.g., the similarity Kðxi, xjÞ > threshold). Papers with similarity less than the threshold will be assigned to disjoint cluster atoms.

Finding cluster atoms would be greatly helpful to name disambiguation. For example, we can take the cluster atoms as the initialization of the disambiguation algorithm. For finding the cluster atoms, one can use a constrained-based clustering algorithm or simply use some constraints. In addition, we define the concept of cluster centroid. Derived from the clustering analysis, there are typically two methods to find the centroid of a

cluster, the data point that is nearest to the center of the cluster or the centroid that is calculated as the arithmetic mean of all data points assigned to the cluster.

2.2.Name Disambiguation

Given a person name a, we denote publications containing the author name a as P ¼ fp1; p2; . . . ; png. The publication data with relationships can be modeled by networks comprising nodes and edges. We use an adaptive version of the so-called informative graph [13] to represent the publication data. Publications and relationships are transformed into an undirected graph, in which each node represents a paper and each edge a relationship. Attributes of a paper are attached to the corresponding node as a feature vector. For the vector, we use words (after stop words filtering and stemming) in the attributes of a paper as features and use the number of their occurrences as the values. Formally, we can define the publication informative graph as follows:

2.2.1.Definition 3 (Publication Informative Graph)

Given a set of papers P ¼ fp1; p2; . . . ; png, let rkðpi; pjÞ be a relationship rk between pi and pj. A publication informative graph is a graph G ¼ ðP; R; VP;WRÞ, where each vðpiÞ 2 VP corresponds to the feature vector of paper pi and wk 2 WR denotes the weight of relationship rk. Let rkðpi; pjÞ ¼ 1 iff there is a relationship rk between pi and pj; otherwise, rkðpi; pjÞ ¼ 0.

Suppose there are K persons fy1; . . . ; yKg with the name a, our task is to disambiguate the n publications to their real researcher yi; i 2 ½1;K_. More specifically, the major tasks of

name disambiguation can be defined as:

- Formalizing the disambiguation problem. The formalization needs to consider both local attribute features associated with each paper and relationships between papers.
- Solving the problem in a principled approach. Based on the formalization, propose a principled approach and solve it in an efficient way.
- Determining the number of people K. Given a disambiguation task (without any prior information), determine the actual K.

It is nontrivial to perform these tasks. First, it is not immediately clear how to formalize the entire disambiguation problem in a unified framework. Second, some

graph models, e.g., Markov Random Field [16], are usually applied to model relational data. However, in the publication informative graph, the papers might be arbitrarily connected by different types of relationships. It is unclear how to perform inference (or parameter estimation) in such a graph with arbitrary structure. In addition, estimating the number of people K is also a challenging task.

3.Parameter Estimation

3.1.Algorithm

The parameter estimation problem is to determine the values of the parameters and to determine assignments of all papers. More accurately, we optimize the log-likelihood objective function (8) with respect to a conditional model At a high level, the learning algorithm (cf. Algorithm 1) for parameter estimation primarily consists of two iterative steps: Assignment of papers, and Update of parameters. The basic idea is that we first randomly choose a parameter setting _ and select a centroid for each cluster.

Next, we assign each paper to its closest cluster and then calculate the centroid of each paper-cluster based on the assignments. After that, we update the weight of each feature function by maximizing the objective function.

Algorithm 1. Parameter estimation

Input: $P=\{p_1, p_2, \ldots, p_n\}$

Output: model parameters Θ and $Y=\{y_1, y_2, \ldots, y_n\}$, where $y_i \in [1, K]$

1. Initialization

1.1 randomly initialize parameters Θ;

1.2 for each paper x_i, choose an initial value y_i, with $y_i \in [1, K]$;

1.3 calculate each paper cluster centroid $\mu_{(i)}$;

1.4 for each paper x_i and each relationship (x_i, x_j), calculate $f_i(y_i, x_i)$ and $f_k(y_i, y_j)$.

2. Assignment

2.1 assign each paper to its closest cluster centroid;

3. Update

3.1 update of each cluster centroid;

3.2 update of the weight for each feature function.

Algorithm 2: One-step sampling
Input: current observation x^0 and labels y^0
Output: sampling results of y^1 and x^1

1: Draw an observation x, from the distribution of $q^0(x_i)$ ($q(x)$ can be obtained by summing over all possible labels);
2: Compute $P(y_i|x)$, the posterior probability distribution over the label variable given the observation x;
3: Compute $P(y_i|y_{\neg i})$, the probability distribution over the label variable given labels of its neighboring observations;
4: Draw a new label y^1_i for each observation from the probability distribution $P(y_i|x)P(y_i|y_{\neg i})$;
5: Given the chosen label, compute the conditional distribution of $P(x_i|y_i)$;
6: Draw each feature of the new observation x^1_i from the conditional distribution $P(x_i|y_i)$.

Finally, based on the reconstructed data vector, we can calculate (13). The stochastic sampling sometimes is time demanding. To make it more efficient, one can use the deterministic mean field algorithm [44] to replace the sampling procedure.

After solving the third term in (10), we can compute the solution for the whole objective function. Finally, a greedy algorithm is used to sequentially update the assignment of each paper. An assignment of a paper is performed while keeping the other papers fixed. The process is repeated until no paper changes its assignment between two successive iterations.

3.2.Estimation Of K

Our strategy for estimating K (see Algorithm 2) is to start by setting it as 1 and we then use the BIC score to measure whether to split the current cluster. The algorithm runs iteratively. In each iteration, we try to split every cluster C into two subclusters. We calculate a local BIC score of the new sub model M2. We calculate a global BIC score for the new model. The process continues by determining if it is possible to split further. Finally, the model with the highest global BIC score is chosen.

Algorithm 3. Estimation of K

Input: $P=\{p_1, p_2, \ldots, p_n\}$

Output: K, $Y=\{y_1, y_2, \ldots, y_n\}$, where $y_i \in [1,K]$

```
1:  i=0, K=1, that is to view P as one cluster: C^(i)={C_1};
2:  do{
3:    foreach cluster C in C^(i){
4:      find a best two sub-clusters model M_2 for C;
5:      if(BIC(M_2)>BIC(M_1))
6:        split cluster C into two sub clusters C^(i+1)={C_1, C_2};
7:      calculate BIC score for the obtained new model;
8:  }while(existing split);
9:  choose the model as output with the highest BIC score;
```

One difficulty in the algorithm might be how to find the best two sub cluster models for the cluster C (Line 4). With different initialization, the resulting sub clusters might be different. Fortunately, this problem is alleviated in our framework, benefiting from the cluster atoms identification. In disambiguation, a cluster can consist of several cluster atoms. To split further, we use the cluster atoms as initializing centroids and thus our algorithm tends to result in stable split results.

4.Conclusion And Future Work

In this paper, we have investigated the problem of name disambiguation. We have formalized the problems in a unified framework and proposed a generalized probabilistic model to the problem. We have defined a disambiguation objective function for the problem and have proposed a two-step parameter estimation algorithm. We have also explored a dynamic approach for estimating the number of people K. Experimental results indicate that the proposed method significantly outperforms the baseline methods. When applied to expert finding, clear improvement (þ2%) can be obtained. As the next step, it would be interesting to investigate how to make use of the time information for name disambiguation, as the ambiguity problem evolves with the time. Moreover, it is also interesting to study how topic models like LDA can improve name disambiguation.

5.Reference

1. H. Akaike, “A New Look at the Statistical Model Identification,” IEEE Trans. Automatic Control, vol. AC-19, no. 6, pp. 716-723, Dec. 1974.
2. S. Basu, M. Bilenko, and R.J. Mooney, “A Probabilistic Framework for Semi-Supervised Clustering,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’04), pp. 59- 68, 2004.
3. R. Bekkerman and A. McCallum, “Disambiguating Web Appearances of People in a Social Network,” Proc. Int’l Conf. World Wide Web (WWW ’05), pp. 463-470, 2005.
4. O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S.E. Whang, and J. Widom, “Swoosh: A Generic Approach to Entity Resolution,” The VLDB J., vol. 18, pp. 255-276, 2008.
5. I. Bhattacharya and L. Getoor, “Collective Entity Resolution in Relational Data,” ACM Trans. Knowledge Discovery from Data, vol. 1, article 5, 2007.
6. C. Buckley and E.M. Voorhees, “Retrieval Evaluation with Incomplete Information,” Proc. Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’04), pp. 25-32, 2004.
7. Z. Chen, D.V. Kalashnikov, and S. Mehrotra, “Adaptive Graphical Approach to Entity Resolution,” Proc. Seventh ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL ’07), pp. 204-213, 2007.
8. Z. Chen, D.V. Kalashnikov, and S. Mehrotra, “Exploiting Context Analysis for Combining Multiple Entity Resolution Systems,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’09), pp. 207-218, 2009.
9. D. Cohn, R. Caruana, and A. McCallum, “Semi-supervised Clustering with User Feedback,” Technical Report TR2003-1892, Cornell Univ., 2003.
10. D. Cai, X. He, and J. Han, “Spectral Regression for Dimensionality Reduction,” technical report, 2856, UIUC 2004.
11. P.T. Davis, D.K. Elson, and J.L. Klavans, “Methods for Precise Named Entity Matching in Digital Collections,” Proc. ACM/IEEECS Joint Conf. Digital Libraries (JCDL ’03), p. 125, 2003.
12. C. Ding, “A Tutorial on Spectral Clustering,” Proc. Int’l Conf. Machine Learning (ICML ’04), 2004.

13. M. Ester, R. Ge, B.J. Gao, Z. Hu, and B. Ben-Moshe, "Joint Cluster Analysis of Attribute Data and Relationship Data: The Connected K-Center Problem," Proc. SIAM Conf. Data Mining (SDM '06), 2006.
14. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. PAMI-6, no. 6, pp. 721-742, Nov. 1984.
15. Z. Ghahramani and M.I. Jordan, "Factorial Hidden Markov Models," Machine Learning, vol. 29, pp. 245-273, 1997.
16. [16] J. Hammersley and P. Clifford, "Markov Fields on Finite Graphs and Lattices," Unpublished manuscript, 1971.
17. H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, "Two Supervised Learning Approaches for Name Disambiguation in Author Citations," Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL '04), pp. 296-305, 2004.
18. H. Han, H. Zha, and C.L. Giles, "Name Disambiguation in Author Citations Using a K-Way Spectral Clustering Method," Proc. ACM/ IEEE Joint Conf. Digital Libraries (JCDL '05), pp. 334-343, 2005.
19. G.E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," J. Neural Computation, vol. 14, pp. 1771- 1800, 2002.
20. L. Jiang, J. Wang, N. An, S. Wang, J. Zhan, and L. Li., "GRAPE: A Graph-Based Framework for Disambiguating People Appearances in Web Search," Proc. Int'l Conf. Data Mining (ICDM '09), pp. 199- 208, 2009.
21. M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An Introduction to Variational Methods for Graphical Models," Learning in Graphical Models, vol. 37, pp. 105-161, 1999.
22. R. Kass and L. Wasserman, "A Reference Bayesian Test for Nested Hypotheses and Its Relationship to the Schwarz Criterion," J. Am. Statistical Assoc., vol. 90, pp. 773-795, 1995.s