Author Disambiguation by Hierarchical Agglomerative Clustering with Adaptive Stopping Criterion

Lei Cen Eduard C. Dragut Luo Si
Computer Science Department
Purdue University, USA
{lcen, edragut, lsi}@purdue.edu

Mourad Ouzzani
Qatar Computing Research Institute
Doha, Qatar
mouzzani@qf.org.qa

ABSTRACT

Entity disambiguation is an important step in many information retrieval applications. This paper proposes new research for entity disambiguation with the focus of name disambiguation in digital libraries. In particular, pairwise similarity is first learned for publications that share the same author name string (ANS) and then a novel Hierarchical Agglomerative Clustering approach with Adaptive Stopping Criterion (HACASC) is proposed to adaptively cluster a set of publications that share a same ANS to individual clusters of publications with different author identities. The HACASC approach utilizes a mixture of kernel ridge regressions to intelligently determine the threshold in clustering. This obtains more appropriate clustering granularity than non-adaptive stopping criterion. We conduct a large scale empirical study with a dataset of more than 2 million publication record pairs to demonstrate the advantage of the proposed HACASC approach.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

Keywords

Author Disambiguation, Clustering

1. INTRODUCTION

The entity resolution problem consists of two subproblems: *disambiguation* and *reference identification.* In the former problem the task is to distinguish references that share the same author name string (ANS) and yet refer to different author identities. For example, there are 13 different author identities sharing the ANS Ashish Garg in DBLP (the Nov. 2012 version) and 7 different authors with the ANS Stefan Richter. The reference identification task determines the set of different ANSs that may be used to refer to the same author identity. For example, Fernando Casadevall, Fernando Casadevall Palacio, Fernando J. Casadevall refer to the same author identity in DBLP. This paper focuses on the disambiguation problem.

Author name disambiguation is an important research problem for bibliographic (Web) databases (e.g., DBLP, CiteSeer, MEDLINE). While substantial efforts are made to clean these repositories by semi-automatic means (which oftentimes goes unrecognized: for instance, DBLP support group utilizes sophisticated heuristic rules to identify ambiguous author names, which are then manually validated [11]), their efforts cannot keep pace with the volume of data ingested in these repositories: These databases are largely constructed by periodically crawling the online proceedings of conferences, workshops and journals. Case in point, DBLP version March 2012 has 671 distinct ambiguous ANSs which are (confidently) disambiguated by the DBLP support group to refer to 2,013 different author identities. A total of 29,103 publications belong to these authors in DBLP. The Nov. 2012 version of DBLP has 143 new ambiguous ANSs that are (confidently) disambiguated, i.e., a 21.3% increase from the previous version. Notice that 88,916 new ANSs and 178,806 new publication records were added to DBLP in Nov. 2012, which were not in DBLP in March 2012. This problem is not unique to DBLP. In MEDLINE, on average 8 different author identities are associated with each ambiguous ANS and 2/3 of the author identities are associated with an ambiguous ANS [13]. This clearly points out that, at such a data ingestion rate, the (admirable) efforts of the curators of DBLP, as well as those of its sister bibliographic repositories, cannot keep pace unless assisted by reliable automated tools.

This paper proposes a novel solution for the author disambiguation problem. Our solution consists of two steps. First, we estimate pairwise similarity between publications sharing the same ANS using Logistic Regression. Second, we use a Hierarchical Agglomerative Clustering (HAC) algorithm to cluster the publications to real author identities. The stopping criterion in HAC is adaptively learned from supervised information.

Our contributions in this paper are:

- Propose a novel method for author disambiguation based on learning adaptive stopping criteria for individual ambiguous ANSs in clustering.
- Conduct a comprehensive large scale empirical study using DBLP, showing that HACASC outperforms HAC with a single fixed threshold as the stopping criterion.

The paper is organized as follows. Section 2 gives a brief overview of the related work. Section 3 describes our pro-

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.

posed solution and Section 4 shows the experimental results. The paper concludes with Section 5.

2. RELATED WORK

There is a rich body of work on the disambiguation problem in general and on the author name disambiguation problem in particular. These problems are part of the more general problem of *entity resolution* (also referred to as record linkage, reference reconciliation, etc.). Several surveys [6, 8] give a thorough presentation on the work on the entity resolution problem. Due to the space limitation, we only review some research work most related to the paper.

A number of solutions have been proposed for the disambiguation problem: unsupervised clustering solutions [13, 14], supervised clustering methods based on naive Bayes and support vector machines [9], graph-based mining, such as, co-authorship graph [7, 12, 2] and entity-relationship graph mining [10], hidden Markov fields [16], and link analysis between publication records using random walks [15].

Our work distinguishes from previous researches on disambiguation problem as we focus on learning adaptive stopping criterion during the clustering process for identifying individual author identities. [2] proposed blocking and boostrapping approach with HAC, but did not elaborate the stopping criterion in clustering. The novel HACASC approach intelligently learns adaptive stopping criterion in clustering, which substantially improves the performance of author disambiguation.

3. METHOD DESCRIPTION

This section first presents a formal definition to the author disambiguation task, and then describes the new method for author disambiguation. The method consists of two main phases. The first phase models the probability that a publication pair sharing an ANS is written by the same author identity. This probability is used as a similarity metric between publications in the second phase, where HACASC is utilized to generate clusters of individual author identities.

3.1 Task Formulation

The mathematical definition of the author disambiguation task is as follows. Let $\mathrm{N} = \{n_1, n_2, \cdots, n_N\}$ be the set of ambiguous ANSs, and $\mathrm{E} = \{e_1, e_2, \cdots, e_M\}$ be the set of real author identities. Each ambiguous ANS $n_i \in \mathrm{N}$ is associated with a set of publications P_{n_i}. For a paper p, denote $Au(p) = \{r_1, r_2, \cdots\}$ as the set of author references in the author list of p, $En(r)$ denotes the real author identity of r, and $Nm(r)$ denotes the ANS of r appearing in the author list. For each author identity $e \in \mathrm{E}$, let $Nm(e)$ be its ANS. The disambiguation problem thus becomes: for each ambiguous ANS n_i, find a partition $C_{n_i} = \{c^1_{n_i}, c^2_{n_i}, \cdots, c^{k_{n_i}}_{n_i}\}$, where $\bigcup_{j=1}^{k_{n_i}} c^j_{n_i} = P_{n_i}$ and $c^j_{n_i} \cap c^k_{n_i} = \emptyset$ if $j \neq k$, such that, $\forall j \in \{1, \cdots, k_{n_i}\}, \exists e \in \mathrm{E}, Nm(e) = n_i, \forall p \in c^j_{n_i}, \exists r \in Au(p), En(r) = e$. For example, let $n \in \mathrm{N}$ be an ambiguous ANS and $P_n = \{p_1, p_2, p_3\}$ the set of publications where n appears. Hence, we have author references $r_1 \in Au(p_1), r_2 \in Au(p_2)$, and $r_3 \in Au(p_3)$ such that $Nm(r_1) = Nm(r_2) = Nm(r_3) = n$. Suppose that $En(r_1) = En(r_2)$ and $En(r_3) \neq En(r_1), En(r_2)$, i.e., r_1 and r_2 refer to the same author identity, which is different from the author identity referred to by r_3. The author disambiguation task is to cluster P_n into two clusters $\{p_1, p_2\}$ and $\{p_3\}$ so that the sets of publications in each cluster correctly indicate the identity of author references r_1, r_2 and r_3.

3.2 Pairwise Similarity Modeling

Let $p1$ and $p2$ be two publications such that $r1 \in Au(p1), r2 \in Au(p2)$, $Nm(r1) = Nm(r2) = n$. To provide a similarity metric for the clustering, the pairwise probability $Pr(En(r1) = En(r2) \,|\, p1, p2)$ is modeled as a Logistic Regression(LR), i.e.

$$Pr(En(r1) = En(r2) \ |p1, p2) = \sigma(w^T \phi(n, p1, p2))$$

where $\sigma(x) = (1 + exp(-x))^{-1}$ is the sigmod function and $\phi(n, p1, p2)$ is the feature vector extracted from $p1$ and $p2$ w.r.t n, which reflects the "similarity" between the two papers for sharing the same real author identity with the ANS n. w is the weight vector indicating the importance of each feature. We will discuss the features used here later in Section 4.2. The learning process of the LR problem is through gradient decent. In particular, the BFGS pseudo Newton method [4] is used to solve this optimization problem.

3.3 Hierarchical Agglomerative Clustering with Adaptive Stopping Criterion

Here we describe the HACASC method for clustering the publications P_n that share an ANS n. There are two issues for this clustering task: first, the number of real author identities that share this ANS is not given, hence the number of clusters is not pre-determined; second, given only the similarity between publications, without a feature vector for each publication, it is hard to compute cluster centers. To overcome the first issue, the natural choice is to use HAC. HAC starts by treating each node as a cluster by itself, and then iteratively merges the closest pair of clusters until some stopping criterion is met. To overcome the second issue, we utilize the following similarity measure between clusters:

$$Sim(c^p_n, c^q_n) = \frac{1}{|c^p_n||c^q_n|} \sum_{\substack{p1 \in c^p_n \\ p2 \in c^q_n}} Pr(En(r1) = En(r2) \ |p1, p2)$$

where $Pr(En(r1) = En(r2) \ |p1, p2)$ is provided by the pairwise similarity modeling (Section 3.2).

An important problem when using the HAC algorithm is how to specify the stopping criterion. A simple choice may be to find a single fixed threshold via training and apply it to future data. Suppose N is partitioned into training set ANSs $\mathrm{N}_{Tr} \subset \mathrm{N}$, and testing set ANSs $\mathrm{N}_{Te} \subset \mathrm{N}$, $\mathrm{N}_{Tr} \cap \mathrm{N}_{Te} = \emptyset$. With the ground truth of the training set, the best threshold t_n, for all $n \in \mathrm{N}_{Tr}$ can be found. Then a single fixed threshold may be determined using these best thresholds in training set(see Section 4.3.2). But using a single fixed threshold for all different ANSs is not optimal. Therefore, this paper proposes new research for adaptively finding the desired thresholds for different ANSs in HAC as a regression problem, i.e. $t_n = f(n, P_n)$. In this regression model, the input sample is a HAC problem with ANS n and related publications P_n, and the target t_n is the best threshold for this HAC problem. With a regression model, the stopping criterion of a HAC problem can be intelligently learned from the optimal stopping thresholds of training samples with known ground truth (i.e., real author identities).

In particular, the regression function f is defined as a mixture of kernel ridge regressions:

$$t_n = \sum_h Pr(Z = h | n, P_n) \sum_{i=1}^{|\mathrm{N}_{Tr}|} \alpha_{i,h} K(n, n_i), \ n_i \in \mathrm{N}_{Tr}$$

	dim.	feature
ANS	2	$IDF_p(F)$, $IDF_p(L)$
	2	$IDF_n(F)$, $IDF_n(L)$
publication title	1	$Sim_{cos\_tfidf}(t1, t2)$
	4	$Sim_{cos\_LDA}(t1, t2)$
co-authorship	2	$CA_1(p1, p2), log(CA_1(p1, p2))$
	2	$CA_2(p1, p2, n), log(CA_2(p1, p2, n))$
venue	1	$Sim_{cos\_tfidf}(v1, v2)$
year	1	$\|y1 - y2\|$

Table 1: Features($\phi(n, p1, p2)$) for pairwise similarity modeling. "dim." stands for feature dimensions.

where Z indicates the hidden group, $Pr(Z = h|n, P_n)$ is the gate function for assigning a HAC task to a hidden group, and $\sum_{i=1}^{|\mathrm{N}_{Tr}|} \alpha_{i,h} K(n, n_i)$ is the kernel ridge regression with $K(\cdot,\cdot)$ as the kernel function. Soft-max function is used for $Pr(Z = h|n, P_n)$ and Radial Basis Function (RBF) [5] kernel for $K(\cdot,\cdot)$.

To learn the mixture of kernel ridge regressions model, the Expectation-Maximization (EM) method is applied. In the E-step, the posterior probability is estimated as follows:

$$Pr(Z = h|n, P_n) = \frac{w_h^T \psi(n, P_n) \mathcal{N}(\sum_{i=1}^{|\mathrm{N}_{Tr}|} \alpha_{i,h} K(n, n_i)|t_n, \beta_h)}{\sum_l w_l^T \psi(n, P_n) \mathcal{N}(\sum_{i=1}^{|\mathrm{N}_{Tr}|} \alpha_{i,l} K(n, n_i)|t_n, \beta_l)}$$

where $\psi(n, P_n)$ is the feature vector, which will be discussed later in Section 4.3.1. $\mathcal{N}(\cdot|t_n, \beta_l)$ is the probability density function of the normal distribution with the best threshold t_n as mean and variance β_l. Here the error term $error = t_n - f(n, P_n)$ is assumed to follow some zero-mean normal distribution.

In the M-step, the parameters to be estimated are $w = \{w_1, \cdots, w_H\}$ for the gate functions, $\alpha = \{\alpha_1, \cdots, \alpha_H\}$ for the kernel ridge regression models in each hidden group and the error term variance $\beta = \{\beta_1, \cdots, \beta_H\}$. The statistics for updating the parameters are:

$$\begin{aligned} w_h^* = \quad & \mathrm{argmax}_{w_h} \sum_{i=i}^{|\mathrm{N}_{Tr}|} \sum_{h=1}^{H} Pr(Z = h|n, P_n) \cdot \\ & log(\frac{1}{Z_{n_i}} exp(w_h^T \psi(n_i, P_{n_i}))) + \lambda' |w_h|^2 \end{aligned}$$

$$\alpha_h^* = D_h(\lambda I_{|\mathrm{N}_{Tr}|} + K D_h)^{-1} T$$

$$\beta_h^* = \frac{\sum_{i=1}^{|\mathrm{N}_{Tr}|} Pr(Z = h|n_i, P_{n_i})(t_{n_i} - \sum_{j=1}^{|\mathrm{N}_{Tr}|} \alpha_{j,h} K(n_i, n_j))^2}{\sum_{i=1}^{|\mathrm{N}_{Tr}|} Pr(Z = h|n_i, P_{n_i})}$$

where $Z_{n_i} = \sum_{h=1}^{H} exp(w_h^T \psi(n_i, P_{n_i}))$ is the normalizer, D_h is the diagonal matrix with $Pr(Z = h|n_i, P_{n_i})$ as the i^{th} diagonal element, K is the kernel matrix of training samples, T is the vector of the best thresholds of all training samples, and λ is the regularization parameter for kernel ridge regression. All the estimations are in closed form except for w_h^*. Again, the BFGS method is used for this optimization problem and another regularization parameter λ' is used to avoid over-fitting. Both regularization parameters, λ for regression model and λ' for gate function are obtained by cross validation in training set.

4. EXPERIMENTAL RESULTS

The goal of the experimental section is to show the advantage of learning adaptive thresholds in the proposed HACASC method. We evaluate the proposed HACASC against the baseline, which uses HAC with a single fixed threshold.

Precision	Recall	F1
0.746	0.843	0.792

Table 2: Performance of LR for the pairwise similarity modeling

4.1 Dataset

We perform our experiments on a subset of DBLP called DBLP_Note dataset. It is compiled from DBLP March 2012. It consists of *all* those ANSs in DBLP with the property that each of them is shared by at least two distinct author identities and each of the author identities has an affiliation note. We consider the presence of affiliation notes as a *strong* indicator that the author identities are "unequivocally" identified by the DBLP support group for those ANSs. DBLP_Note consists of 692 ambiguous ANSs, of which 354 ANSs are used for training and 338 ANSs are used for testing. By pairing up the publications of the authors in DBLP_Note that share the same ANS, there are 1,109,733 pairs from 15,394 publications in the training set and 1,027,641 pairs from 14,578 publications in the testing set.

4.2 Pairwise Similarity Model Experiments

We report here the experimental results for the first phase of our approach. Recall that a LR model is built to model the pairwise similarity between publications sharing an ANS.

4.2.1 Feature Extraction

Table 1 shows the features used for the LR model. Two name-based features $(IDF_p(F), IDF_p(L))$ calculate the Inverse Document Frequency(IDF) of the first(last) names of the given ANSs against all publications in the whole DBLP, i.e. $log(\frac{\text{\#pub.}}{\text{\#pub. w/ ANS w/ the first name}})$. Another two features $(IDF_n(F), IDF_n(L))$ compute IDF for the first(last) names against all ANSs in DBLP , i.e. $log(\frac{\text{\#ANS}}{\text{\#ANS with the first name}})$. One title-based feature uses cosine similarity with TF-IDF features (Sim_{cos_tfidf}) of publication titles and another four use Latent Dirichlet Allocation (LDA) [3] features (Sim_{cos_LDA}) instead. To compute the LDA features, a LDA model is first built using all the publication titles in the training set. It is then applied to publication titles. The estimated topic assignment probabilities of the titles are denoted as the LDA features of the titles. LDA models with hidden group sizes $10, 30, 50$ and 80 are used to generate the four features.

For co-authorship features, the level-1 and level-2 co-authorship similarity are defined as follows:

$$CA_i(p1, p2, n) = \sum_{n' \in Co_i(p1,p2,n)} log(\frac{\text{\#author name}}{\text{\#co-author name of } n'})$$

where $i \in \{1, 2\}$, and $Co_1(p1, p2, n)$ is the set of ANSs appearing in both $p1$ and $p2$ besides n, $Co_2(p1, p2, n)$ is the set of ANSs that appear in $p1(p2)$ and has co-authorship with some ANS in $p2(p1)$ besides n. Here the co-authorship is based on ambiguous ANSs, not real author identities, so it is not the accurate co-authorship.

Finally, a *venue* feature is computed using cosine similarity and TF-IDF features; the *year* feature is computed as the absolute value of the difference of the publication years.

4.2.2 Modeling Performance

Table 2 shows the precision, recall and F1 score of the learned LR model. The metrics here are computed by taken the pairwise similarity modeling as a classification problem. A threshold is selected in training set to truncate the simi-

Method	F1	NMI
baseline	0.810	0.422
HACASC	0.832†	0.544†
UpperBound	0.927	0.739

Table 3: Clustering Performance Comparison

larity into a binary number which is then compared to the ground truth of whether a pair shares the same author identity. Notice that this is only a pairwise result, and may contain conflicts. E.g. the model may predict that both $\{p1, p2\}$ and $\{p2, p3\}$ share the same author identity e, but $\{p1, p3\}$ does not.

4.3 Experimental Results for HACASC

In the experiments for HACASC, we first describe the features used for the HACASC, then compare the performance between usage of adaptive threshold and a single fixed threshold.

4.3.1 *Feature Extraction*

Table 4 shows the features used for the HACASC, where $S = \{Pr(En(r1) = En(r2)\ |p1, p2)\ |p1, p2 \in P_{n_i}\}$ and $V = \{\sum_{p2 \neq p1, p2 \in P_{n_i}} Pr(En(r1) = En(r2)\ |p1, p2)\ |p1 \in P_{n_i}\}$. The name features are the same as in LR. The pairwise similarity features show the average of the pairwise similarity between the publications sharing the same ANS. The node volume features show the density of the complete graph consisting of related publications and their similarities.

4.3.2 *Clustering Performance*

Here we evaluate the proposed HACASC method against a baseline and a theoretical upper bound. The baseline uses a single threshold as a weighted sum of the best thresholds in the training set, with the sizes of HAC problems ($|P_{n_i}|$) as weights. The theoretical upper bound is the performance using the best threshold gained from ground truth for each ANS. The evaluation metric includes F1 score and Normalized Mutual Information (NMI) [1]. Unlike the result in pairwise modeling, the F1 score is derived from the clustering result here, hence the transitive conflicts mentioned in Section 4.2.2 do not apply here. The NMI is used to evaluate the performance from the information-theoretic interpretation of clustering, while F1 score evaluates the performance from the pairwise perspective of clustering, as series of decisions. The NMI is computed as a weighted (with the sizes of HAC as weights) sum of the NMIs of each of the HAC problems w.r.t. the correct clustering results (given by the ground truth). Table 3 shows the clustering performance. The RBF kernel used in HACASC has one scale parameter, tuned using cross-validation. The number of hidden groups is 5, which in our experiments performs much better than < 5 groups and similar to > 5 groups.

It can be seen from Table 3 that the HACASC generates a better F1 and much better NMI score in testing set compared to the baseline. To confirm this, a right-tailed t-test is applied for both F1 and NMI with statistical significance 99.9% ($\alpha = 0.1\%$). The resulting p-value is 3.81×10^{-19} for F1 score and 1.97×10^{-32} for NMI, indicating substantial advantage of HACASC against the baseline. The upper bound performances show very good pairwise results (over 90% F1 score), which mean that the pairwise modeling does a good job in ranking the publication pairs, but the thresholds are very different for different HAC problems.

	dim.	feature
ANS	2	$IDF_p(F)$, $IDF_p(L)$
	2	$IDF_n(F)$, $IDF_n(L)$
pairwise similarity	2	$mean(S), std(S)$
node volume	2	$mean(V), std(V)$

Table 4: Features ($\psi(n, P_n)$) for the regression in HACASC

5. CONCLUSION AND FUTURE WORK

This paper proposes a HACASC method to intelligently determine the threshold in a HAC process for the author disambiguation problem. This method utilizes Logistic Regression to model the pairwise publication similarity, and the mixture of kernel ridge regressions to model the adaptive thresholds for the stopping criteria of the HAC problems. Our experiments in DBLP_Note dataset show substantial advantage of HACASC against the baseline, in both classification and information-theoretic perspective. There is still a large difference between the performance of the upper bound and HACASC. One possible improvement is to incorporate the supervised information with the unsupervised information, such as within cluster distance and between cluster distance, to determine the stopping criterion, which may result in a more effective model.

Acknowledgments: This work is partially supported by NSF research grants IIS-0746830, CNS-1012208, IIS-1017837 and IIS-0916614. It is also partially supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

6. REFERENCES

[1] R. Balasubramanyan, F. Lin, and W. Cohen. Node clustering in graphs: An empirical study. In *WNADTA*, 2010.

[2] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *TKDD*, 1(1), Mar. 2007.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] M. D. Buhmann. *Radial basis functions: theory and implementations*. Cambridge university press, 2003.

[6] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *TKDE*, 2007.

[7] X. Fan, J. Wang, X. Pu, L. Zhou, and B. Lv. On graph-based name disambiguation. *JDIQ*, 2(2), 2011.

[8] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender. A brief survey of automatic methods for author name disambiguation. *SIGMOD Rec.*, 2012.

[9] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *JCDL*, 2004.

[10] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *TODS*, 31(2), June 2006.

[11] M. Ley. DBLP: some lessons learned. *PVLDB*, 2009.

[12] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel. Co-authorship networks in the digital library research community. *Inf. Process. Manage.*, 2005.

[13] V. I. Torvik and N. R. Smalheiser. Author name disambiguation in medline. *TKDD*, 3(3), 2009.

[14] X. Yin, J. Han, and P. S. Yu. Object distinction: Distinguishing objects with identical names. In *ICDE*, 2007.

[15] X. Yin, J. Han, P. S. Yu, and I. T. J. Watson. Object distinction: Distinguishing objects with identical names by link analysis. In *ICDE*, 2007.

[16] D. Zhang, J. Tang, J. Li, and K. Wang. A constraint-based probabilistic framework for name disambiguation. In *CIKM*, pages 1019–1022, 2007.