Efficient Name Disambiguation for Large-Scale Databases

Jian Huang[1], Seyda Ertekin[2], and C. Lee Giles[1,2]

[1] College of Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802, U.S.A.
{jhuang, giles}@ist.psu.edu
[2] Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802, U.S.A.
sertekin@cse.psu.edu

Abstract. Name disambiguation can occur when one is seeking a list of publications of an author who has used different name variations and when there are multiple other authors with the same name. We present an efficient integrative framework for solving the name disambiguation problem: a blocking method retrieves candidate classes of authors with similar names and a clustering method, DBSCAN, clusters papers by author. The distance metric between papers used in DBSCAN is calculated by an online active selection support vector machine algorithm (LASVM), yielding a simpler model, lower test errors and faster prediction time than a standard SVM. We prove that by recasting transitivity as density reachability in DBSCAN, transitivity is guaranteed for core points. For evaluation, we manually annotated 3,355 papers yielding 490 authors and achieved 90.6% pairwise-F1. For scalability, authors in the entire CiteSeer dataset, over 700,000 papers, were readily disambiguated.

1 Introduction

Name disambiguation is desired in many cases: e.g., evaluating faculty publications, calculating statistics of social network and author impacts, etc. The metadata of publications such as authors, titles etc. is very valuable for automatic bibliometrics and citation analysis. Manual extraction of metadata can be costly for large-scale digital libraries such as the Google Scholar and CiteSeer. Automatic metadata extraction [1] is not perfect especially for papers crawled from the web, where many items are missing or incomplete. With author profiles constructed from disambiguation, these fields can be correctly populated, improving the quality of existing metadata.

Name disambiguation is an interesting data mining problem with AKA's and other pseudonyms. The problem is deemed challenging in large-scale digital libraries. First, name disambiguation is a *meta* problem. Unlike disambiguation in NLP, name disambiguation in academic papers does not necessarily have context in a document, since authors do not appear in the text. In our case we use the metadata of an author's papers to determine his identity. Moreover, *scalability* is a significant concern for large-scale databases, thus giving a preference

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): PKDD 2006, LNAI 4213, pp. 536–544, 2006.

for unsupervised or semi-supervised methods since it's implausible to annotate and train a classifier for each namesake. In addition, *expandability* is an issue for persistent disambiguation. As new papers come in, more information is available to refine previous results and name clusters could be adjusted when appropriate.

Our **contribution** is addressing the above challenges as follows:

- We use an online SVM algorithm (LASVM) to build a supervised distance function, which yields a simpler yet faster model with active learning.
- We overcome the transitivity problem commonly found in other disambiguation work by using an efficient clustering algorithm DBSCAN.
- Our framework is easily expandable to new papers: the supervised learner for the distance function can easily handle additional data with online learning; also, DBSCAN can adjust name clusters based on the new information.
- The framework integrates supervised and unsupervised methods to provide a scalable solution, and is readily amendable to various improvements.

2 Related Work

Prior name disambiguation work mainly deals with the **citation matching** problem [2, 3, 4]. Hybrid Naive Bayes and Support Vector Machine [5] methods are inappropriate for large-scale databases, due to the cost of human annotation. K-spectral clustering was used in [4] to find an approximation of the global optimal solution. However, the computation complexity $O(N^2)$ is intractable for large-scale databases. Also, K is unknown *a priori* for an increasing database. The scalability issue is addressed in [6] by using a two-level blocking framework, reducing computation complexity to $O(C\,|B|)$ (C is the number of blocks and $|B|$ the average size of blocks). However, citations are differentiated by single pairwise distance without clustering. Like earlier agglomerative clustering approaches [7], this could lead to the **transitivity problem**, due to the noisy data and the inaccurate distance function. Multiple distances instead of a threshold on single pairs are accounted for in [8], imposing transitivity by adding an additional feature (with weight $-\infty$) into the Conditional Random Field model. We ameliorate the problem by using an efficient unsupervised clustering method DBSCAN [9], which also makes coreferent decisions based on multiple distances.

3 Methods

3.1 Solution Overview

We formalize *name disambiguation* in Fig.1 as:

> Given a research paper $p^{(i)}$, each author appearance $a_u^{(i)}$ in this paper is associated with a metadata record $r_u^{(i)}$, consisting of a set of attributes $\{t_{u,k}^{(i)}\}_{k=1}^{m}$. Our goal is to find an assignment function Θ, such that $\Theta(a_u^{(i)}) = E_w$, where E_w represents the real entity; in other words, $\Theta(a_u^{(i)}) = \Theta(a_v^{(j)})$ if and only if $a_u^{(i)}$ and $a_v^{(j)}$ refer to the same person.

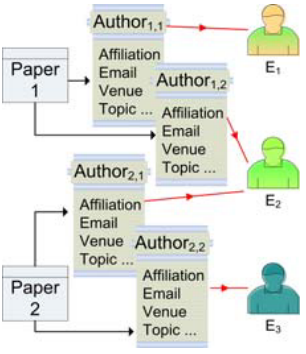

Fig. 1. Author disambiguation

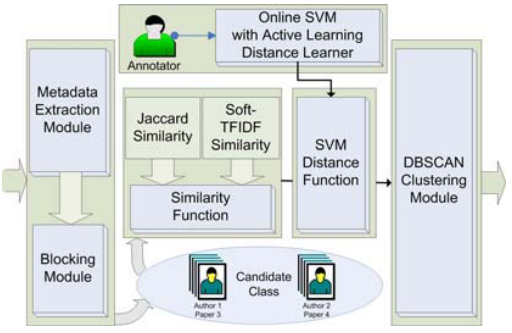

Fig. 2. Disambiguation system overview

Fig.2 shows the system architecture. The metadata extraction module [1] first extracts author metadata records from each paper. The blocking module then blocks namesakes into candidate classes including only non-conflicting name variations, thus significantly reduces the number of similarity calculation for pairs from the entire database to within candidate classes. Afterward, the similarity function computes a similarity vector $s^{(i,j)}=[sim_1(t^{(i)}_{u,1},t^{(j)}_{v,1}),...,sim_m(t^{(i)}_{u,m},t^{(j)}_{v,m})]^T$, from the attributes $\{t^{(i)}_{u,k}\}^m_{k=1}$ and $\{t^{(j)}_{v,k}\}^m_{k=1}$ in record pairs, corresponding to author appearances $a^{(i)}_u$ and $a^{(j)}_v$ in a candidate class. We use different similarity predicates sim_l depending on the nature of the attributes. For instance, the edit distance is used for emails and URLs; token-based Jaccard similarity for addresses and affiliations; hybrid similarity Soft-TFIDF [10] for name variations.

The SVM then uses the similarity vector $s^{(i,j)}$ as a feature vector to classify whether $r^{(i)}_u$ and $r^{(j)}_v$ are coreferent, and the confidence of coreference is used as a pairwise distance metric. Finally, DBSCAN constructs clusters based on multiple pairwise distances, which addresses the transitivity problem. These last two modules are described in more detail in the rest of this section.

3.2 Distance Function with Online SVM and Active Learning

In the hypothetical space $\mathcal{R}$ spanned by metadata records, we need to determine the distance $dist(s^{(i,j)})$ between two records $r^{(i)}_u$ and $r^{(j)}_v$. The distance function $dist$ is non-trivial and data-driven, thus we use a supervised learning algorithm to determine such a function. Support Vector Machine (SVM) [11] is originally designed for binary classification and shows good generalization performance. We, however, use the SVM's to obtain the learner's confidence in the coreferent class as in [10]. The confidence values determine the distances between the record pairs, i.e., the more confident the SVM model classifies two metadata records as coreferent, the closer they are in $\mathcal{R}$. For simplicity of notation, we refer to the training sample $s^{(i,j)}$ as $\mathbf{x_k}$ and its true label as y_k. Given a labeled training dataset $\{(\mathbf{x_1}, y_1), ..., (\mathbf{x_N}, y_N)\}$ $(y_i \in \{-1, +1\})$, the SVM aims to find

an 'optimal' hyperplane $(\mathbf{w} \cdot \mathbf{x}) + b = 0$ ($\mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}$) that separates the training data, after solving the optimization problem of minimizing the function $\mathbf{L}(\mathbf{w}) = \|\mathbf{w}\|^2/2$, subject to $y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1$ ($i = 1, ..., N$).

We efficiently train the SVM model with an online kernel classifier LASVM [12]. LASVM relies on the traditional soft margin SVM formulation, while it works faster and preserves the classification accuracy rates of the state-of-the-art traditional SVM solvers. Traditional SVM works in a batch setting; whereas LASVM works in an online setting, where its model is continually modified as new training instances become available. The speed improvement and the less memory demand with **online learning** makes LASVM applicable to very large datasets. When the digital library is populated with new papers, LASVM can integrate the information of the new data without retraining all the samples, thus it is adaptable to growing datasets.

In our setting, the metadata records are inherently noisy, thus not all the training samples are equally informative. We believe that by using only the most informative samples and discarding the noisy samples, we will get a simpler and sparser model. This can be accomplished by **active sample selection**. In SVM based active learning, the most informative sample among all the training data is the one closest to the hyperplane. Classical active learning method with SVM's [13] is computationally expensive as it requires a search through all the unseen training samples. We use a method as in [12] that will not necessitate a full search, but locates an approximate most informative sample by examining a small constant number of randomly chosen samples. The method first picks M ($M = 50$ as in [12]) random training samples and selects the best one among them. Thus active sample selection can be done in reasonable time.

3.3 DBSCAN Clustering

We use the clustering method DBSCAN [9] to cluster author appearances in papers and prove that it can handle the inconsistency of author labeling. We do not simply classify whether two metadata records are coreferent or not simply based on the pairwise distance $dist(s^{(i,j)})$, due to the **transitivity problem**: for a triad (o, p, q), point o is coreferent with p, and p with q, while o is not coreferent with q. This is an inconsistent condition since coreference should be transitive, and is due to errors in metadata extraction, imperfect similarity metric and misclassification. We formally prove that DBSCAN for the most cases resolves the transitivity problem. Other reasons that we choose DBSCAN include:

- Minimal domain knowledge is needed to determine the two parameters ϵ and MinPts, which can be tuned by visualization methods such as OPTICS [14].
- DBSCAN can model clusters of any arbitrary shape and delimit clusters more intuitively to human interpretation.
- DBSCAN is highly efficient, for its computation complexity is $O(N \log N)$.

We briefly review the definitions in DBSCAN that are used in our theorems. For an author appearance $a_u^{(i)}$ in a candidate class $\mathcal{D}$, DBSCAN induces the cluster $\mathcal{C}$ from $\mathcal{D}$ such that $\forall a \in \mathcal{C}, \Theta(a) = E_w$.

Definition 1. *Point p is* ***directly density reachable*** *from q, if* $p \in N_\epsilon(q)$ *and* $|N_\epsilon(q)| \geq MinPts$*(core point condition), where* $N_\epsilon(q) = \{p \in \mathcal{D} \,|dist(p,q) \leq \epsilon\}$.

Definition 2. *Point p is* ***density reachable*** *from q if there exists a chain* $p_1 = p, ..., p_n = q$*, such that* p_{i+1} *is directly density reachable from* p_i.

Definition 3. *Point p is* ***density connected*** *to q if there exists o, such that both p and q are density reachable from o.*

Definition 4. *A cluster* $\mathcal{C}$ *is a subset of* $\mathcal{D}$ *satisfying:*
1.(Maximality) $\forall p, q$*, if* $p \in \mathcal{C}$ *and q is density reachable from p, then* $q \in \mathcal{C}$.
2.(Connectivity) $\forall p, q \in \mathcal{C}$*, p is density connected to q.*

Under the DBSCAN framework, we recast the **coreference** relationship as **density connectivity**, both of which are symmetric. Formally speaking, for all $p, q \in \mathcal{D}$, p is coreferent with q iff p is density connected to q. We prove the following theorem which shows that the transitivity problem with DBSCAN is no longer an issue for the most part.

Theorem 1. *Transitivity is guaranteed as long as p is a core point.*

Proof. A contradiction exists if the transitivity problem exists, i.e., o is not coreferent with q. o is coreferent with p implies that o is density connected to p, so there exists r such that p and o are density-reachable from r. Hence two chains $a_1 = r, ..., a_k = o$ and $b_1 = r, ..., b_l = p$ exist, such that a_{i+1} and b_{i+1} are directly density reachable from a_i and b_i respectively. Specifically, $p \in N_\epsilon(b_{l-1})$ implies $b_{l-1} \in N_\epsilon(p)$. Since p is a core point, we have $|N_\epsilon(p)| \geq MinPts$. Thus b_{l-1} is directly density reachable from p. Note that by definition of density reachable, $b_1 ... b_{l-1}$ should all satisfy core point condition. This forms a reverse chain $b_l = p, ..., b_1 = r$ such that b_{i-1} is directly density reachable from b_i. Now we have formed a density reachable chain from p to o ($b_l = p, ..., b_1 = a_1 = r, ..., a_k = o$), and similarly another chain from p to q. Thus o is density connected to q, which violates the assumption that o is not coreferent with q. □

Theorem 2 determines the correctness of DBSCAN for coreference resolution, and corollary 1 dictates the absence of transitivity problem within a cluster.

Theorem 2. $\forall \mathcal{C}$ *and* $\forall p, q \in \mathcal{C}$*, p is coreferent with q.*

Proof. By connectivity property in definition 4, $\forall p, q \in \mathcal{C}$, p is density connected to q. Therefore, p is coreferent with q.

Corollary 1. *The transitivity problem does not exist for any triad in a cluster.*

Combining Theorem 1 and Corollary 1, we are left with the case where the transitivity problem exists: p is a border point ($|N_\epsilon(q)| < MinPts$) of different clusters. The nature of density-based clustering implies that this is a rare case since such points will lie on the cluster boundary and will be sparse. Such points are due to insufficient information which would be necessary to disambiguate a

Table 1. Author datasets (R=#records, A=# authors)

ID	Dataset	R	A
1	A. Gupta	506	44
2	A. Kumar	143	36
3	C. Chen	536	103
4	D. Johnson	350	41
5	J. Anderson	327	43
6	J. Robinson	115	30
7	J. Smith	743	86
8	K. Tanaka	53	20
9	M. Jones	352	53
10	M. Miller	230	34
	Total	3,355	490

Table 2. SVM models testing results: LASVM vs. LIBSVM

ID	Error(%)		Prediction Time(sec.)[a]	
	LIBSVM	LASVM(%chg.)	LIBSVM	LASVM(%chg.)
1	19.34	**17.989**(-7.00%)	137.3	**109.3(-20.4%)**
2	6.491	**6.149**(-5.26%)	6.3	**5.1(-19.0%)**
3	**4.882**	4.885(+0.07%)	118.8	**94.2(-20.7%)**
6	2.814	**2.335**(-17.0%)	5.3	**4.1(-22.6%)**
7	9.721	**9.168**(-5.69%)	215.6	**170.2(-21.1%)**
8	11.00	**10.513**(-4.45%)	1.1	**0.8(-27.3%)**
10	21.31	**18.35(-13.9%)**	25.3	**19.6(-22.5%)**
Avg	11.218	**9.913(-7.60%)**	72.8	**57.6(-23.5%)**

[a] Test on Dell Precision 370 server (3.0GHz Xeon CPU)

particular person's name on a paper. When more information is available, the problem can be easily solved with DBSCAN by merging or splitting clusters.

To sum up, by using the SVM to learn the underlying distance function, DBSCAN acts as an assignment function Θ to disambiguate authors in papers.

4 Experiments

We empirically study the efficiency and effectiveness of our proposed method by testing both the supervised distance function and the entire framework. Using the CiteSeer metadata (obtained from SVM-based metadata extraction [1]), 10 most ambiguous names are sampled from the entire dataset as listed in Table 1. These names are in parallel with the names used in [5, 4] representing the worst case scenario, and are geographically diverse to cover names of different origins. 3,355 papers are manually labeled yielding 490 authors. For those ambiguous author names from different papers, we meticulously went through the original papers, homepages, CVs, etc, to confirm their authorship.

4.1 Experiments on SVM Based Distance Function

We select datasets with ID number 4, 5 and 9 as a three-fold training dataset, consisting of 81,073 pairwise coreference training samples. Our first goal is to obtain a simpler model for efficient distance calculation. As we see in Fig. 3, in active learning setups, after using certain number of training data, the number of support vectors saturates and the test error stabilizes. We observe that adding more training data after this point hardly changes the model. This implies that the most informative samples are already included in the model and the remaining samples do not provide extra information. Therefore, we determine an early stopping point for training by cross validation results (Fig. 3). We first select an interval of iteration number from 12,310 to 14,100, where the average cross validation error is stably minimized. Then we fix the iteration number to 14,100, where the number of support vectors is closest to saturation.

Our LASVM model is trained on the entire training dataset, stopping the training process at this iteration number. For comparison, we also train a classical SVM model with a popular implementation LIBSVM [15] using batch learning. Table 2 shows the test error and prediction time of LASVM, compared to classical SVM, for the seven test datasets. Our model demonstrates 23.5% reduction in the prediction time on average, due to the decrease in the number of support vectors from 9,822 to 7,809. This simpler model also achieves 7.6% decrease in test error, implying a more accurate distance function.

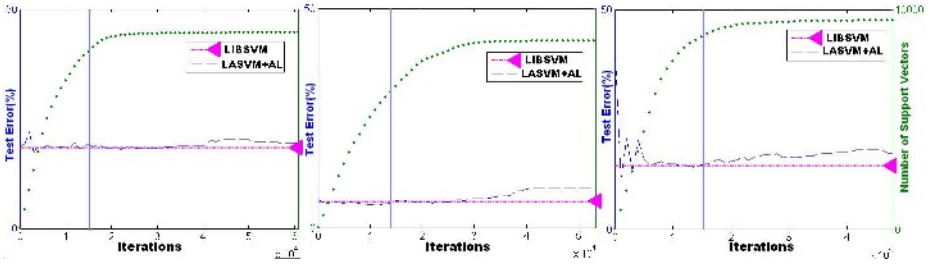

Fig. 3. Cross-validation on three-fold training datasets (from left to right: train[4,5] test[9]; train[4,9] test[5]; train[5,9] test[4]). The optimal iteration number for early stopping is shown with a vertical line, and the LIBSVM test error with a triangle.

4.2 Name Disambiguation Performance

We measure disambiguation performance at two levels as in [3]. At the pair level, **pairwise precision** pP is defined as the fraction of pairs in the same cluster being coreferent, **pairwise recall** pR as the fraction of coreferent pairs put into the same cluster, and **pairwise F1** $pF1$ as the harmonic mean of pP and pR. At the cluster level, **cluster precision** cP is the ratio of the number of completely correct clusters to the total number of clusters retrieved, whereas **cluster recall** cR is the portion of true clusters retrieved. Likewise, **cluster F1** $cF1$ is the harmonic mean. The **ratio of cluster size** RCS is defined as the number of clusters retrieved versus the number of true clusters. Note that cluster level metrics give no credits to clusters that miss some papers or are partially correct, making them more stringent and less telling than the pairwise metrics.

Table 3 shows the disambiguation accuracy of the entire system. Overall, it achieves 90.6% pairwise F1 metric, and 63.8% of the author name clusters are completely correct. The RCS is 0.944 (close to the optimal value 1.0), implying that the number of unique authors can be estimated with the number of clusters from disambiguation results. To test the efficiency, the entire CiteSeer metadata dataset is disambiguated in 3,880 minutes, yielding 418,809 unique authors.

5 Conclusion

An integrative framework is introduced to efficiently and adaptively resolve the name disambiguation problem. In this framework, a blocking module significantly

reduces the cost of similarity calculation. Our results show that with active sample selection and early stopping, learning a distance function is faster and more accurate than that of traditional SVM's. Our framework is easily expandable to the growing datasets. First, online setting enables the incorporation of new information without retraining the entire collection. Second, DBSCAN corrects the rare cases where the transitivity property is violated by merging or splitting clusters. We also formally prove the correctness of using DBSCAN for coreference resolution and the absence of transitivity problem for core points.

Table 3. Disambiguation accuracy

Dataset	pP	pR	pF1	cF1	RCS	Dataset	pP	pR	pF1	cF1	RCS
A. Gupta	0.914	0.960	0.937	0.483	0.977	J. Smith	0.815	0.853	0.834	0.625	0.860
A. Kumar	0.995	0.941	0.972	0.667	0.845	K. Tanaka	0.980	1.000	0.990	0.923	0.950
C. Chen	0.782	0.970	0.866	0.739	1.049	M. Jones	0.895	0.873	0.884	0.717	0.774
D. Johnson	0.761	0.948	0.844	0.434	1.024	M. Miller	0.775	0.953	0.855	0.451	1.028
J. Anderson	0.909	0.978	0.942	0.675	0.791	**Mean**	0.873	0.944	0.906	0.638	0.944
J. Robinson	0.908	0.963	0.935	0.667	1.143	**Std. Dev.**	0.085	0.046	0.056	0.150	0.122

Acknowledgments. This work was partially supported by grants from Microsoft Research and the National Science Foundation (NSF).

References

[1] Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.: Automatic document metadata extraction using support vector machines. In: Proceedings of Joint Conference on Digital Libraries (JCDL 2003). (2003) 37–48

[2] McCallum, A., Nigam, K., Ungar, L.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of KDD. (2000)

[3] Wellner, B., McCallum, A., Peng, F., Hay, M.: An integrated, conditional model of information extraction and coreference with application to citation matching. In: Proceedings of the 20th Conference on Uncertainty in AI. (2004) 593–601

[4] Han, H., Zha, H., Giles, C.L.: Name disambiguation in author citations using a K-way spectral clustering method. In: Proceedings of JCDL. (2005) 334–343

[5] Han, H., Giles, C.L., Zha, H., Li, C., Tsioutsiouliklis, K.: Two supervised learning approaches for name disambiguation in author citations. In: Proceedings of Joint Conference on Digital Libraries (JCDL 2004). (2004) 296–305

[6] Lee, D., On, B., Kang, J., Park, S.: Effective and scalable solutions for mixed and split citation problems in digital libraries. In: ACM SIGMOD Workshop on Information Quality in Information Systems (IQIS). (2005)

[7] Mann, G.S., Yarowsky, D.: Unsupervised personal name disambiguation. In: Proceedings of CoNLL-7. (2003) 33–40

[8] Bekkerman, R., McCallum, A.: Toward conditional models of identity uncertainty with application to proper noun coreference. In: IJCAI Workshop. (2003)

[9] Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996). (1996) 226–231

[10] Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.: Adaptive name-matching in information integration. IEEE Intelligent System **18**(5) (2003)
[11] Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
[12] Bordes, A., Ertekin, S., Weston, J., Bottou, L.: Fast kernel classifiers with online and active learning. Journal of Machine Learning Research **6** (2005) 1579–1619
[13] Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: Proc. of 7th International Conf. on Machine Learning (ICML). (2000)
[14] Ankerst, M., Breunig, M., Kriegel, H., Sander, J.: OPTICS: Ordering points to identify the clustering structure. In: Proc. of ACM SIGMOD. (1999) 49–60
[15] Chang, C., Lin, C.: LIBSVM: a library for support vector machines. (2001)