A Latent Dirichlet Model for Unsupervised Entity Resolution

Indrajit Bhattacharya Lise Getoor
Department of Computer Science
University of Maryland, College Park, MD 20742

Abstract

Entity resolution has received considerable attention in recent years. Given many references to underlying entities, the goal is to predict which references correspond to the same entity. We show how to extend the Latent Dirichlet Allocation model for this task and propose a probabilistic model for collective entity resolution for relational domains where references are connected to each other. Our approach differs from other recently proposed entity resolution approaches in that it is a) generative, b) does not make pair-wise decisions and c) captures relations between entities through a hidden group variable. We propose a novel sampling algorithm for collective entity resolution which is unsupervised and also takes entity relations into account. Additionally, we do not assume the domain of entities to be known and show how to infer the number of entities from the data. We demonstrate the utility and practicality of our relational entity resolution approach for author resolution in two real-world bibliographic datasets. In addition, we present preliminary results on characterizing conditions under which relational information is useful.

1 Introduction

In many applications, there are a variety of ways of referring to the same underlying entity. Given a collection of entity references, or references for short, we would like to a) determine the collection of 'true' underlying entities and b) correctly map the references in the collection to these entities. This problem comes up in many guises throughout computer science. Examples include computer vision, where we need to figure out when regions in two different images refer to the same underlying object (the correspondence problem); natural language processing where we would like to determine which noun phrases refer to the same underlying entity (co-reference resolution); and databases, where, when merging two databases or cleaning a database, we need to determine when two records are referring to the same underlying individual (deduplication).

We are interested in resolving references when they are connected to each other via relational links, as in the bibliographic domain where author names in papers are connected by co-author links. Now entity resolution becomes collective in that resolution decisions depend on each other through the relational links. We show that collective entity resolution improves performance over independent pair-wise resolution.

There is a long history of work in both general and relational entity resolution. Recently, generative [22, 29] and discriminative [24, 28] probabilistic approaches have been proposed as well as non-probabilistic algorithms [20, 12]. Our model differs from most of the above in that it is unsupervised, does not assume the underlying entities to be known, does not make pairwise decisions and explicitly models relations between entities using group membership.

We introduce a generative probabilistic model for entity resolution that builds on the recently proposed Latent Dirichlet Allocation model (LDA) [6]. Unlike most existing models, we do not introduce a decision variable for each potential duplicate pair of references, but instead have an entity label for each reference. To model collaborative relations between entities, we introduce a group label for each reference, so that entities coming from the same collaborative group are more likely to be observed in a relation. For author resolution, this means that we model collaborative groups to explain co-authorship relations. The generative process in our model may be viewed as an extension of the Dirichlet Process mixture model: the group labels in our model influence the choice of entities for each author reference in a paper.

Another contribution of this paper is an unsupervised Gibbs sampling algorithm for collective entity resolution. It is unsupervised because we do not make use of a labeled training set and it is collective because the resolution decisions depend on each other through the group labels. Further, the number of entities is not fixed in our model, and we propose a novel sampling strategy to estimate the most likely number of entities given the references.

The paper is organized as follows. We present a motivating example in Section 2 and related research in Section 3. In Section 4, we first adapt the LDA model for document authors and extend it for entity resolution in Section 5. The sampling framework for inference is presented in Section 6. In Section 7 and Section 8, we describe how entity attributes are modeled. Section 9 describes our novel algorithm for determining the number of entities and in Section 10 and Section 11 we explore parameter choices and algorithmic improvements . Finally, we present experimental results on real and synthetic data in Section 12 and conclude in Section 13.

2 A Motivating Example

In this section, we introduce a concrete bibliographic example to explain the entity resolution problem for authors and motivate our approach. Consider as an example six real paper citations P1 through P6 from CiteSeer:

P1: "JOSTLE: Partitioning of Unstructured Meshes for Massively Parallel Machines" C. Walshaw, M. Cross, M. G. Everett, S. Johnson

P2: "Partitioning Mapping of Unstructured Meshes to Parallel Machine Topologies", C. Walshaw, M. Cross, M. G. Everett, S. Johnson, K. McManus

P3: "Dynamic Mesh Partitioning: A Unified Optimisation and Load-Balancing Algorithm", C. Walshaw, M. Cross, M. G. Everett

P4: "Code Generation for Machines with Multiregister Operations", Alfred V. Aho, Stephen C. Johnson, Jefferey D. Ullman

P5: "Deterministic Parsing of Ambiguous Grammars", A. V. Aho, S. C. Johnson, J. D. Ullman

P6: "Compilers: Principles, Techniques, and Tools", A. Aho, R. Sethi, J. Ullman

Each of the 6 papers has its own author references. For instance, the first paper P1 has four references 'C. Walshaw', 'M. Cross', 'M. G. Everett' and 'S. Johnson'. In all we have 21 references in the 6 papers. The goal is to find out how many different author entities these references correspond to and which reference maps to which entity. Ground truth tells us that all of the Aho's map to the same author entity, as do the Everret's and the Ullman's. The interesting case here is that of Johnson. The four Johnson references correspond to two Johnson entities: those in papers P4 and P5 correspond to Stephen C. Johnson from Bell Labs, while those in papers P1 and P2 map to Steve P. Johnson from University of Greenwich, London. However, going by just the names of the references it is not clear why 'Stephen C. Johnson' is not 'S. Johnson', when 'Alfred V. Aho' is the same as 'A. Aho'. Our goal will be to make use of the collaboration relationships to make these contrasting inferences simultaneously. We would like to be able to infer from the collaborations that there are two different collaboration groups in this example and authors are more likely to publish with other authors from the same group. As illustrated in Fig. 1, the first group G1 has Aho, Ullman and Sethi as member authors. The other group G2 has Walshaw, Cross, Everett and McManus. Stephen C. Johnson is associated with the first collaboration group, while S Johnson from papers P1 and P2 is a different person since he is associated with the second collaboration group.

In order to make these inferences, our model introduces an entity label and a group label for each reference, both of which are hidden and need to be inferred. The inference procedure is collective in that they cannot be made independently for each reference — their relationships to other references need to be considered as well. Also, the group and the entity labels are inter-dependent. The entity labels for

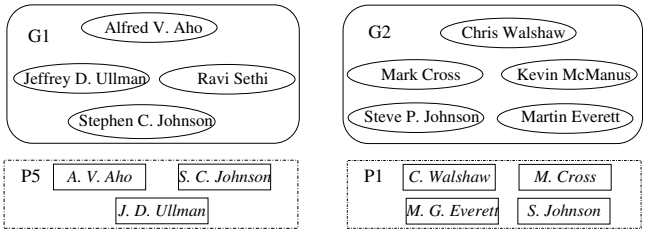

Figure 1: Author entities in two different collaboration groups and two generated papers. The ovals are the entities belonging to groups shown as encapsulating rectangles. Dotted rectangles represent papers with author references shown as smaller solid rectangles. Each paper is generated by the group above it.

the two Johnson's depend on their group labels, as we just saw. Also, the group labels depend on the entity labels in turn. Sethi from paper P6 and Johnson from paper P5 belong to the same group since they are tied by the identical entity labels for the Aho's and Ullman's in the two papers. These two hidden variables are the key distinctions of our model in comparison to some other recent ones that have been proposed. Most other approaches introduce a decision variable for each potential duplicate pair to infer whether or not they correspond to the same entity, while we introduce two variables for each reference in the data. As data sizes grow, we believe that this distinction has a significant impact.

It is interesting to note the role of papers P3 and P6 in this collective inference for the Johnson's though none of them contain a Johnson reference. They help to reinforce our belief that there are two distinct tightly knit groups or communities where member authors collaborate strongly with each other. Observe that frequent collaborations between Walshaw and Aho, and Everett and Ullman for example would have the opposite effect. Then we would think there is one collaboration group, as opposed to two, and therefore all Johnson's are more likely to be the same author.

Not surprisingly, inferring the entity labels exactly turns out to be intractable. In this paper, we propose an effective Gibbs sampling approach for approximate inference. Also, one critical aspect of the inference procedure is discovering the likely number of entity labels, since the actual entities are hidden from us. We show how the number of entities can be inferred as well.

Though we use the bibliographic domain of papers and authors, our model is applicable in a straight-forward manner for other domains where noisy references to person entities are observed together. Examples include names of people traveling together on the same flight, names appearing together in the same email or groups of people attending the same meeting. Furthermore, our approach can be general-

ized to model other resolution problems. We are investigating a very similar model for word sense resolution in natural language documents, where the references are word occurrences and the senses are the entities to be resolved.

3 Related Work

There is a large body of work on deduplication, record linkage, and co-reference detection. The traditional approach to entity resolution considers similarity of textual attributes. There has been extensive work on approximate string matching algorithms [26, 8] and adaptive algorithms that learn string similarity measures [4, 9, 33]. Beyond applying standard machine learning techniques, other approaches use active learning [32]. In addition, data integration is an area of active research [17, 26, 23].

The groundwork for posing record linkage as a probabilistic classification problem was done by Fellegi and Sunter [13]. Winkler [34] builds upon this work by introducing a latent match variable estimated using Expectation Maximization. More recently, hierarchical graphical models have been proposed [30].

Probabilistic models that take into account interaction between different entity resolution decisions have been proposed for named entity recognition in natural language processing and for citation matching. McCallum et al. [24] employ conditional random fields (CRF) for noun coreference and use clique templates with tied parameters where the decision for one pair affects another through their overlap. Parag et al. [28] extend the CRF model to merge evidence across multiple fields. More recently, Culotta and McCallum [10] have considered relations between multiple types to deduplicate them jointly. However, all of these models consider pairwise decisions between potential duplicates and are supervised in that their parameters require training on labeled data. Our approach is distinct in that the parameters do not require training and are estimated automatically from unlabeled data. Also, we do not consider pairwise decisions which becomes prohibitive for bigger datasets. Instead, we use an entity label for each reference.

Pasula et al. [29] propose a probabilistic relational model for the citation matching problem. This captures dependence between identities of co-authors of the same paper, but does not model collaborative probabilities between authors directly. Daumé and Marcu [19] have recently proposed an extension to Pasula et al.'s model, where the number of clusters or entities is directly modeled by a Dirichlet Process and is similar in spirit to ours. However, we propose a three level model where the selection of author entities depends on the groups that they belong to. Li et al. [22] propose a generative model for disambiguating entities in text documents that captures joint probabilities for co-occurrence. They show impressive benefits over a pairwise discriminative model. They model pairwise co-occurrence probabilities rather than group memberships and searching for the set of most likely entities is not a focus of their work.

Kalashnikov et al. [20] enhance feature-based similarity between an ambiguous reference and the many entity choices for it with relationship analysis between the entities, like affiliation and co-authorship. This is in the same spirit as our work, however they focus on the entity matching problem where the domain of entities is given and the right entity needs to be identified for each new reference. We focus on a more difficult problem where neither the entities nor the number of entities is known.

Non-probabilistic approaches that take relational features into account for data integration have been proposed [11, 7, 1, 3, 20, 12]. Chaudhuri et al. [7] make use of join information for deduplication but assume the secondary tables themselves to be clean. The notion of co-occurrence in dimensional hierarchies has also been proposed [1], while other approaches look at weighted combinations of attribute and relational distance measures [3]. Dong et al. [12] adopt a model similar to Parag et al. [28] and resolve entities of multiple types by propagating relational evidences in a dependency graph. They adopt a pair-wise reconciliation approach so that the graph has nodes for all potential duplicate pairs and all pairs of similar attributes.

We model collaborative groups using LDA [6] which improves Probabilistic Latent Semantic Indexing [18] as a generative topic model for documents. The related author-topic model [31] recognizes the problem of duplicate authors; here we propose a solution for it. Kubica et al. [21] have proposed generative models for links using underlying groups, but they do not handle identity uncertainty.

4 LDA Model for Authors

In this section, we show how the LDA model for topics and words in documents can be adapted to a group mixture model for author entities. We start with the simpler case where there is no ambiguity in the author references. In the next section, we expand the model to handle ambiguous author references.

Consider a collection of D documents and a set of A authors who write these documents. We have a set of R author references, $\{a_1, \ldots, a_R\}$ in these D documents. Each document can have multiple authors and for now, we assume the authors of each document are observed. For the i^{th} author reference, a_i indicates which author it corresponds to and d_i denotes the document in which it occurs. Further we introduce the notion of collaborative author groups. These are groups of authors which tend to co-author together. We will assume that there are T different groups. Each author reference a_i has an associated group label z_i.

The probabilistic model is shown using plate notation in Figure 2(a). The probability distribution over authors for each group is represented as a multinomial with parameters ϕ^j, so the probability $P(a = i \mid z = j)$ of the i^{th} author in

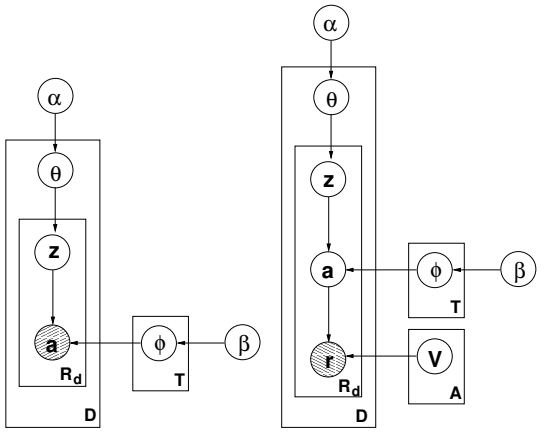

Figure 2: Plate representation for (a) LDA model for authors and (b) LDA-ER model for author resolution from ambiguous references. Observed variables are shaded.

the database being chosen for the j^{th} group is ϕ_i^j. We have T different multinomials, one for each group. Each paper d is modeled as a mixture over T groups. The distribution used is again a multinomial with parameters θ^d, so the probability $P_d(z = j)$ of the j^{th} group being chosen for document d is θ_j^d. Each θ^d is drawn from a Dirichlet distribution with hyper-parameters α; similarly each ϕ^j is drawn from a Dirichlet distribution with hyper-parameters β.

To illustrate this generative process in the model, we show how the authors for paper P5 are chosen in Fig. 1. First, a distribution θ^d over collaborative groups is chosen for the paper. These are the likely groups that will contribute the authors of the document. Each group has a distribution ϕ^i over likely authors. In our example, ϕ^{G1} has equal probability for Aho, Ullman, Sethi and Stephen C. Johnson and 0 otherwise, while ϕ^{G2} chooses between Walshaw, Cross, Everett, Steve Johnson and McManus with equal probability. Note that our model allows an author to belong to multiple groups, though not illustrated here. The distribution θ^d that is chosen for paper P1 has probability 1 for group $G1$ and 0 probability for all other groups. Now each author is chosen by first sampling a group z_i from θ^d and then sampling an author from group z_i. Since θ^d for P1 has non-zero probability only for group $G1$, it is the group that is chosen for every author in P1. Having selected $G1$ as the group for each author, the first draw from ϕ^{G1} yields Aho as the first author, the second yields Stephen C Johnson and the third yields Ullman. The authors for the other papers are selected similarly. Note that in general more than one group may have non-zero probability in the distribution θ^d for a paper, so that authors for the same paper can come from multiple groups with smaller probability.

5 LDA-ER Model for Author Resolution

In the previous section, we assumed that the author identity can be determined unambiguously from each author reference. However, when we are dealing with author names, this is typically not the case. The same author may be represented in a variety of ways: 'Alfred V. Aho', 'Alfred Aho', 'AV Aho', etc. There may be mistakes due to typos or extraction errors. Finally, two 'S. Johnson's may not refer to the same author entity. One may refer to 'Stephen C. Johnson' and another may refer to 'Steve P. Johnson'. The result is that we are no longer sure of the mapping from the author reference to the author entity. We must resort to inference to identify the true author for each reference.

To capture this, we will associate an attribute v_a with each author a. In addition, we add an extra level to the model that probabilistically modifies the author attributes V_a to generate the references $\mathbf{r} = \{r_1, r_2, \ldots, r_R\}$. Each reference is generated by first sampling a group z and then an author entity a as before. Then, the author reference r is generated from a by modifying the attribute v_a according to a noise model $\mathcal{N}$. We use a relatively sophisticated noise model that we explain in Section 8. The probability of generating an author reference r from a particular author entity is defined as $P(r|v_a)$. The conditional probabilities for each reference are normalized to sum to 1 over all author entities. It is the reference r that is observed, while the entity a and group label z are hidden variables. The LDA-ER model is represented in Figure 2(b).

Illustrating this in the context of our motivating example in Fig. 1, we have already seen how the three author entities are chosen for paper P1. The attributes v_a for the three authors are 'Alfred V. Aho', 'Stephen C. Johnson' and 'Jeffrey D. Ullman'. However the complete/correct names do not always appear in papers or citations. In this case, the noise process modifies the attributes of the three selected entities to generate 'A. V. Aho', 'S. C. Johnson' and 'J. D. Ullman' as the three author references in the paper.

The probability of generating the set $\mathbf{r}$ of references for a corpus given parameters α, β and $\mathbf{V}$ can be expressed as

$$
\begin{aligned}
(5.1)\, P(\mathbf{r}; \alpha, \beta, \mathbf{V}) &= \prod_d P(\mathbf{r}_d; \alpha, \beta, \mathbf{V}) \\
&= \prod_d \sum_{\mathbf{a}_d} P(\mathbf{r}_d \mid \mathbf{a}_d; \mathbf{V}) P(\mathbf{a}_d; \alpha, \beta) \\
&= \int_\phi P(\phi; \beta) \prod_d \sum_{\mathbf{a}_d} P(\mathbf{r}_d \mid \mathbf{a}_d; \mathbf{V}) \\
&\quad \times \int_\theta P(\theta; \alpha) P(\mathbf{a}_d \mid \theta, \phi) d\theta d\phi
\end{aligned}
$$

6 Inference using Gibbs Sampling

In general, the integral in Eq. (5.1) is intractable due to coupling between θ and ϕ. Different approximations have

been proposed, including variational methods [6], Gibbs sampling [16] and Expectation Propagation [25].

We extend the approach proposed by Griffiths et al. [16] for our model. Now θ and ϕ are not directly estimated as parameters. Instead, we first construct the posterior distribution $P(\mathbf{z}, \mathbf{a} \mid \mathbf{r})$ and then estimate θ and ϕ from this posterior distribution. We derive the joint probability from Eq. (5.1) as:

$$P(\mathbf{z}, \mathbf{a}, \mathbf{r}) = P(\mathbf{z})P(\mathbf{a} \mid \mathbf{z})P(\mathbf{r} \mid \mathbf{a}) \tag{6.2}$$

where

$$P(\mathbf{z}) = (\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T})^D \prod_{d=1}^{D} \frac{\prod_t \Gamma(\alpha + C_{dt}^{DT})}{\Gamma(T\alpha + C_{d*}^{DT})} \tag{6.3}$$

is the probability of the joint group assignment to all references and

$$P(\mathbf{a} \mid \mathbf{z}) = (\frac{\Gamma(A\beta)}{\Gamma(\beta)^A})^T \prod_{t=1}^{T} \frac{\prod_a \Gamma(\beta + C_{at}^{AT})}{\Gamma(A\beta + C_{*t}^{AT})} \tag{6.4}$$

is the conditional probability of the authors given the groups and $P(\mathbf{r} \mid \mathbf{a}) = \prod_i P(r_i \mid v_{a_i})$ is the conditional probability of the references given the authors. C_{dt}^{DT} is the number of times group t has been observed for all the references in document d and $C_{d*}^{DT} = \sum_t C_{dt}^{DT}$. Similarly, C_{at}^{AT} is the number of times references to author a have been observed with group label t in all documents.

We construct a Markov chain that converges to the posterior distribution $P(\mathbf{z}, \mathbf{a} \mid \mathbf{r})$ and then draw samples from this Markov chain. Each state in the Markov chain is an assignment of a group label and an author label to all R references. In the Gibbs sampling approach, the labels for each reference are sequentially sampled conditioned on the current labels of all other references. By construction, this Markov chain converges to the target posterior distribution. However, we first need to define the full conditional distribution $P(z_i = t, a_i = a \mid \mathbf{z}_{-i}, \mathbf{a}_{-i}, \mathbf{r})$, where $\mathbf{z}_{-i}$ is the set of all but the i^{th} group label and $\mathbf{a}_{-i}$ of all but the i^{th} author label. In words, this is the probability that the i^{th} reference comes from the t^{th} group and the a^{th} author considering the current group and author assignment to *all other* references.

We can derive this full conditional distribution as

$$\begin{aligned} & P(z_i = t, a_i = a \mid \mathbf{z}_{-i}, \mathbf{a}_{-i}, \mathbf{r}) \\ \propto \; & \frac{C_{(-i)d_i t}^{DT} + \alpha}{C_{(-i)d_i *}^{DT} + T\alpha} \frac{C_{(-i)at}^{AT} + \beta}{C_{(-i)*t}^{AT} + A\beta} P(r_i \mid v_a) \end{aligned}$$

The factorization makes intuitive sense. The first term is the probability of group t in document d_i, the second is the probability of author a in group t and the third is the probability of the author attribute v_a being modified into the i^{th} reference.

Instead of sampling z_i and a_i as a block, we can sample them separately:

$$\begin{aligned} & P(z_i = t \mid \mathbf{z}_{-i}, \mathbf{a}, \mathbf{r}) \\ \propto \; & \frac{C_{(-i)d_i t}^{DT} + \alpha}{C_{(-i)d_i *}^{DT} + T\alpha} \frac{C_{(-i)a_i t}^{AT} + \beta}{C_{(-i)*t}^{AT} + A\beta} \end{aligned} \tag{6.5}$$

$$P(a_i = a \mid \mathbf{z}, \mathbf{a}_{-i}, \mathbf{r}) \propto \frac{C_{(-i)at_i}^{AT} + \beta}{C_{(-i)*t_i}^{AT} + A\beta} P(r_i \mid v_a) \tag{6.6}$$

7 Modeling Author Attributes

In the previous section, while the author labels were unobserved, we assumed that the author attribute values v_a are known. But in general, the author attributes will not be known and we now show how to infer their values from the references. The conditional distribution for sampling groups z_i is not directly affected by the attributes. However, the attributes influence the assignment of author labels a_i, since a reference r_i is more likely to be assigned to an author with similar attributes. Conversely, any author attribute v_i depends on the references that have author label i. Incorporating a prior $P(\mathbf{v}) = \prod_{i=1}^{A} P(v_i)$ into the joint distribution in Eq. (6.2), we derive the conditional distribution for assigning a value v to v_i given all author labels and references as:

$$P(v_i = v \mid \mathbf{a}, \mathbf{r}) \propto P(v) \prod_{j=1}^{R} P(r_j \mid v)\delta_i(a_j) \tag{7.7}$$

Intuitively, v_i should be set to the *most likely* value that explains the generation of the references assigned to author i. For example, if multiple "A.V. Aho" and "Alfred Aho" references have been assigned author label i along with the reference "Alfred Ah", then the author attribute v_i is most likely to be "Alfred V. Aho". The sampling algorithm now also samples the author attributes v_i iteratively, conditioned on the references and current author assignments, along with sampling the group and entity labels for each reference. For 'free authors' to which no references are currently assigned, their attributes cannot be estimated. They are assigned a 'free' attribute '$\star$', that is equally likely to generate any reference attribute.

8 Noise Model

The different ways of distorting or modifying an author attribute to an author reference in a paper is captured by the noise model $\mathcal{N}$. The noise model handles first, middle and last names independently. The first name can be initialed with probability p_{FI}, dropped with probability p_{FD} or retained as a whole with probability p_{FR}, where $p_{FI} + p_{FD} + p_{FR} = 1$. There are similar parameters p_{MI}, p_{MD} and p_{MR} for the middle name. The probabilities for the first and middle initials being incorrect are p_{FIr} and p_{MIr}. Last names

and retained first or middle names may be corrupted by characters being inserted, deleted or replaced with probabilities p_I, p_D and p_R respectively. The minimum numbers of insertion (n_I), deletion (n_D)and replacement (n_R) operations for modifying an author attribute v to a reference v' are obtained using edit-distance for strings. Then the generation probability is $P(v'|v) = p_I^{n_I} \cdot p_D^{n_D} \cdot p_R^{n_R}$.

9 Determining Number of Entities

In the development up until now, we have considered the number of authors A to be given, when in practice this needs to be estimated. One of the contributions of this work is an unsupervised method for determining the number of entities. We propose a novel approach that avoids searching explicitly over the possible number of author entities and instead adapts it within our sampling framework.

9.1 Basic Inference With Gibbs Sampling We first describe a novel but simple Gibbs sampling algorithm for iteratively sampling the values of the hidden group and entity labels for each reference conditioned on the existing labels of all other references. Equations 6.5, 6.6 and 7.7 form the basis of this algorithm. We first sample a group label for each reference according to Eq. (6.5). Next, we sample an entity label for each reference according to Eq. (6.6). The difference for the entities is that the number of entity labels is unknown and needs to be inferred by the algorithm. So we either choose an existing entity label or alternatively a hitherto unused one. For a new entity label, its observed occurrence count $C^{AT}_{(-i)at_i}$ is 0. But the parameter β ensures a non-zero probability of a new label being chosen. Also, the attribute v_a for a new entity is unknown. So we use a fixed value for the probability $P(r_i|v_a)$ for a new entity a that controls how frequently new entity labels are sampled. Once all the entity labels are sampled, in the third step the attribute values are sampled for each of the existing entities according to Eq. (7.7). The iterations continue till convergence. There is a connection between this flavor of Gibbs sampling inference for number of entities and the Dirichlet process which we describe in the next subsection.

9.2 Relation to the Dirichlet Process The Dirichlet process was introduced by Ferguson [14] and Antoniak [2] as a non-parametric statistical approach that allows the complexity of the model to grow with increasing size of the data. In the context of our application, we would like the number of entities to be inferred in model rather than it being a fixed parameter, and we would like the model to be able to accommodate a greater number of entities as the number of references in the data grows. The Dirichlet process can be imagined as a distribution over discrete distributions and is used as follows for choosing the number of components in a mixture model. A distribution (or a component) is first drawn from the Dirichlet process, the parameters are then sampled from this distribution and finally the data is drawn using these parameters. Drawing a parallel with our application, we can sample an entity first, choose the parameters (the attribute) for that entity and then finally generate the reference using the entity parameters. When the Dirichlet process is integrated out, a clustering effect is observed in the conditional distribution for choosing the n^{th} component given $n-1$ previous component draws. The probability of choosing one of the existing components is proportional to the number of times it has been chosen in the previous $n-1$ draws, while a new component has a nonzero probability of being sampled. In particular, let G_0 be the baseline probability distribution over discrete components η and α be a scalar. Then, given the $n-1$ draws $\eta_{1:n-1}$, the distribution for the n^{th} component is given by

$$\eta_n = \begin{cases} \eta_i^* & \text{with prob} \quad \frac{n_i}{n-1+\alpha} \\ \eta, \eta \sim G_0 & \text{with prob} \quad \frac{\alpha}{n-1+\alpha} \end{cases}$$

where n_i is the number of times η_i^* has occurred in $\eta_{1:n-1}$.

Exact inference is intractable in the Dirichlet process mixture model but approximate inference techniques have been proposed [27, 5]. Of particular interest is the Gibbs sampling strategy proposed by Neal [27]. This algorithm iteratively samples the component label a_i for the i^{th} data object r_i from the conditional distribution given the other labels:

$$\begin{aligned} (9.8) \qquad & P(a_i = k \mid \mathbf{r}, \mathbf{a_{-i}}, \alpha) \\ = \; & P(a_i = k \mid \mathbf{a_{-i}}, \alpha) P(r_i \mid \mathbf{r_{-i}}, \mathbf{a_{-i}}, a_i = k) \end{aligned}$$

For an existing component k

$$(9.9) \qquad P(a_i = k \mid \mathbf{a}_{-i}, \alpha) = \frac{C^A_{(-i)k}}{\alpha + N - 1}$$

where $C^A_{(-i)k}$ is the number of previous assignments to the k^{th} component without counting the i^{th} assignment. For a component k that has not been used before

$$(9.10) \qquad P(a_i = k \mid \mathbf{a}_{-i}, \alpha) = \frac{\alpha}{\alpha + N - 1}$$

We may imagine LDA-ER as the Dirichlet process mixture model augmented with a group structure above it that enables it to capture relations between the components or entities. In LDA-ER, a group $z_i = t$ is first sampled for the i^{th} reference from the distribution over groups for the document and then an entity is sampled from it. In the Dirichlet process, any previously existing entity may be chosen in this step depending on their prior counts. But in LDA-ER, the choice is controlled by the sampled group t. Entities that have previously been associated with this sampled group are much more likely to be chosen.

This distinction allows LDA-ER to model relations between entities. As in the Dirichlet process, alternatively a new entity may be selected in LDA-ER. However, this new entity now becomes associated with group t and may be chosen for future references from this group. This difference is clearly observable from the conditional distributions in Eq. (9.9) and Eq. (6.6). While the probability for choosing the k^{th} entity in Eq. (9.9) depends on $C^{A}_{(-i)a}$ which is the number of previous occurrences of entity a, in Eq. (6.6) it depends on $C^{AT}_{(-i)at}$ which is the number of joint occurrences of group t and entity a. This coupling of the group and entity labels distinguishes the LDA-ER model from the Dirichlet process mixture model.

9.3 Block Assignment for Entity Resolution As has been noted in the case of naive Gibbs sampling for inference in the Dirichlet process mixture model [5], iteratively estimating the group and entity label for each reference separately, as described in Sec. 9.1 can be prohibitively slow. We now describe a novel algorithm that overcomes this problem by reassigning entity labels for a set of entities at the same time. This achieves an agglomerative clustering effect on the references. Observe that for any assignment of entity labels to references, each entity label defines a cluster — all references that have this entity label belong to this cluster. Sampling a new label for each reference separately is equivalent to an individual reference migrating from one cluster to another. Agglomerative clustering is significantly faster since pairs of clusters merge into one. We achieve the same effect with the new sampling algorithm that we propose. In addition, we allow existing clusters to split. The conditional probabilities for these choices for any particular entity cluster given the entity and group labels for all other references are derived from the joint distribution in Eq. (6.2). As in traditional Gibbs sampling, these probabilities then form the transition probabilities in a Markov process.

We define a cluster by picking an author label j and consider the set $\mathbf{s}$ of reference indices that have j as their author label: $\mathbf{s} = \{i \mid a_i = j\}$. We assign new author labels to all references indexed by cluster $\mathbf{s}$ simultaneously. In general, the number of possible author assignments to $\mathbf{s}$ is exponential in $|\mathbf{s}|$ and it is virtually impossible to enumerate all these different probabilities and sample from this distribution.

Instead, in our algorithm we restrict the space of candidates such that the cluster of references assigned to a particular author label may (a) merge with a cluster currently assigned to another author label, (b) stay unchanged or (c) split and have a part assigned to a hitherto unassigned author label j'. Case (a) is similar to two author clusters merging and the number of authors is effectively decreased by one. In case (c), an author cluster splits into two and the number of authors is effectively increased by one. However, the number of possible partitions of $\mathbf{s}$ into j and j' is still $2^{|\mathbf{s}|}$. The simple but restricted solution that we use is splitting to the set that last merged into label j via option (a).

We first consider assigning a single author label to all of cluster $\mathbf{s}$. The full conditional distribution we need to derive is $P(\mathbf{a_s} = i \mid \mathbf{z}, \mathbf{a_{-s}}, \mathbf{r})$ which is the probability of all the labels $\mathbf{a_s}$ in cluster $\mathbf{s}$ being set to i conditioned on all references and group labels and all *other* author labels. Let us denote

$$T(t,i) = \prod_{n=1}^{C^{AT}_{(\mathbf{s})it}} (\beta + C^{AT}_{(-\mathbf{s})it} + C^{AT}_{(\mathbf{s})it} - n) \tag{9.11}$$

$$T(t,*) = \prod_{n=1}^{C^{AT}_{(\mathbf{s})*t}} (A\beta + C^{AT}_{(-\mathbf{s})*t} + C^{AT}_{(\mathbf{s})*t} - n)$$

where $C^{AT}_{(\mathbf{s})at}$ is the number of times author a and group t have been jointly assigned to references in $\mathbf{s}$, and $C^{AT}_{(-\mathbf{s})at}$ is the number of such assignments outside $\mathbf{s}$. Let $\mathbf{z_s}$ be the set of groups currently assigned to the references indexed by cluster $\mathbf{s}$. Then the conditional distribution is derived from Eq. (6.2) as

$$\begin{aligned} & P(\mathbf{a_s} = i \mid \mathbf{z}, \mathbf{a_{-s}}, \mathbf{r}) \\ \propto & \prod_{t \in \mathbf{z_s}} \frac{T(t,i)}{T(t,*)} \prod_{j \in s} P(r_j \mid v_i) \end{aligned} \tag{9.12}$$

where the first product term is the group evidence for the assignment and the second is the attribute evidence.

An Interpretation of Block Assignment: Here we show how the terms in this conditional probability can be rearranged so that the result makes intuitive sense. Let j be an index into cluster $\mathbf{s}$ and t_j be the group label for that reference. Also, consider cluster $\mathbf{s}$ to be an ordered set and denote by $\mathbf{s}_{<j}$ the set of elements in $\mathbf{s}$ strictly before position j. Then we can rewrite Eq. (9.13) as

$$\begin{aligned} & P(\mathbf{a_s} = i \mid \mathbf{z}, \mathbf{a_{-s}}, \mathbf{r}) \\ \propto & \prod_{j \in s} \frac{\beta + C^{AT}_{(\mathbf{s}_{<j})it_j} + C^{AT}_{(-\mathbf{s})it_j}}{A\beta + C^{AT}_{(\mathbf{s}_{<j})*t_j} + C^{AT}_{(-\mathbf{s})*t_j}} P(r_j|v_i) \end{aligned} \tag{9.13}$$

Here $C^{AT}_{(\mathbf{s}_{<j})it}$ is the number of times author label i and group label t have occurred jointly for just the references in $\mathbf{s}_{<j}$. We interpret this as follows. We assign author labels to the references in cluster $\mathbf{s}$ in sequence. For each assignment, the second term is the probability of the reference given the author and the first term is the probability of the author label for the reference given its current group label, *including the assignments already made in the sequence as additional evidence*. It must be stressed that this ordering is introduced solely for interpretation purposes and the actual probability

is independent of the ordering. Note that Eq. (9.13) reduces to Eq. (6.6) as expected when cluster $\mathbf{s}$ has a single element.

For the case when we partition cluster $\mathbf{s}$ into $\mathbf{s_1}$ and $\mathbf{s_2}$ and assign two different author labels to them, the conditional probability looks very similar:

$$P(\mathbf{a}_{\mathbf{s_1}} = i, \mathbf{a}_{\mathbf{s_2}} = i' \mid \mathbf{z}, \mathbf{a}_{-\mathbf{s}}, \mathbf{r}) \propto \prod_{t \in \mathbf{z_s}} \frac{T(t,i)T(t,i')}{T(t,*)} \prod_{j \in s_1} P(r_j \mid v_i) \prod_{j \in s_2} P(r_j \mid v_{i'})$$

Observe that when one author label merges with another according to Eq. (9.13), the attribute of the freed author j changes from v_j to the free attribute '$\star$'. The difference in prior probabilities of the two attribute values leads to an additional term in the merge probability in Eq. (9.13): $P(\star)/P(v_j)$. Similarly, when splitting the references assigned to author j between j and currently unassigned j', the attribute of author j' changes to $v_{j'}$ from '$\star$' and the split probability has the additional term $P(v_{j'})/P(\star)$. Therefore, the higher the prior probability of '$\star$' relative to other attributes, the higher will be the likelihood of a merge compared to a split.

Putting everything together, our entity resolution algorithm starts from an initial assignment of authors and groups to all references and iterates over three steps sequentially until convergence. First, it samples a group label for each reference. This has complexity $O(RT)$ for R references and T group labels. Then for each assigned author label, it samples the next author label for its current references. This requires $O(AS)$ operations for A author labels and a maximum of S potential duplicates per author. Finally, it samples an attribute for each assigned author label, requiring $O(A)$ operations. For each round of sampling authors and attributes, we do several iterations of group sampling to let the group labels stabilize for the current author assignments. Note that all stages in an iteration are linear in the number of references and author labels allowing our model to scale to large datasets as we demonstrate in the experimental section.

10 Determining Model Parameters

We have described how the numbers of authors can be determined within the sampling procedure. The remaining aspects of the model are the number of groups and the Dirichlet hyper-parameters. Their choice affects performance in different ways.

10.1 Number of Groups We begin by observing that the choice of the number of groups is subjective and not as critical as the number of entities. Relationships among the same set of entities can be captured with different number of groups at different levels of resolution. While it is possible to estimate the likely number of groups from the data, it is an area of potential future research. Here we consider the effect of varying number of groups on entity resolution. Recall that our guiding intuition is to assign the same author label to sets of references when they are similar *and* have similar group distributions. When the number of groups T is too small, misleading similarities in group distributions are likely to be observed, leading to false positives. If T is too high, references to the same author can get split over different groups, making false negatives likely. In other words, lower T favors higher recall and lower precision, while higher T leads to lower recall with higher precision.

10.2 Hyper-parameters To appreciate the roles of α and β, note from Eq. (6.5) that when $\alpha = 0$, a reference is forced to pick a group label from the other references in the same document. Similarly, when $\beta = 0$, a reference has to pick a group label from other references to the same author, and also an author label from other references with the same group label. In general, for low values of α and β, the model tends to overfit the data. This is particularly undesirable for entity resolution, since we need to estimate the number of authors and need to generalize from the current author assignments. To get a feel for what values are appropriate, observe that $T\alpha$ is the number of pseudo reference counts added to each document. Since in most cases documents will have one or two authors, we set $T\alpha$ to be 0.25. Similarly, $A\beta$ is the number of pseudo references for each topic. We set β according to the number of references in the dataset and the number of topics used. A typical value for $A\beta$ is 5.

10.3 Noise Model Parameters We iteratively estimate the noise parameters from data in a unsupervised manner. We start from an initial estimate that is typical of some datasets we explored. For instance, first names are initialed and dropped with probabilities 0.75 and 0.001 (0.25 and 0.7 for middle names) and is incorrect with probability 0.0005 (0.001 for middle names). Characters may be dropped, replaced or inserted, each with probability 0.0025. After every author sampling step, we re-estimate the probabilities looking at each reference attribute and the attribute of the author it has been assigned to. However, the estimates from the initial iterations may not be good. For example, when all references are distinct entities, all corruption probabilities are estimated to be 0. To prevent this, estimates are made to evolve slowly. A weighted combination of the current probabilities and the new estimates yields the probabilities for the next iteration. Typically, we retain current estimates with weight 0.9.

11 Algorithm Refinements

Unlike group labels, author labels for references are sampled from a restricted space. Here we propose improvements for the sampling algorithm for inferring the author labels.

11.1 Bootstrapping Author Labels Initialization of author labels is an issue both for convergence time and quality. One option is to assign the same initial label to any two references that have attributes v_1 and v_2, where either $v_1 = v_2$ or v_1 is an initialed form of v_2. However, for domains where last names repeat very frequently, like Chinese, Japanese or Indian names, this can affect the initial accuracy quite adversely, from which it is hard to recover. For the case of such common last names[1], we propose an improved bootstrapping scheme. We assign the same author label to pairs only when they have document co-authors with the same initial author label. This improves bootstrap accuracy significantly for one of our datasets that has frequently repeating names.

11.2 Group Evidence for Author Self Loops Recall that Eq. (9.11) shows the group evidence for different transitions for cluster $\mathbf{s}$. $C^{AT}_{(-\mathbf{s})at}$ is the number of references outside cluster $\mathbf{s}$ that have author label a and group label t. For any group t, it is the group evidence for merging with the cluster for author label a. However, if $\mathbf{s}$ is the cluster of references with author level j, then $C^{AT}_{(-\mathbf{s})jt}$ will be 0 for all group labels t, since there are no references outside cluster $\mathbf{s}$ with author label j. Therefore, cluster $\mathbf{s}$ has little affinity to itself when considering group evidence and prefers merging with other clusters. Note however that every cluster has higher attribute affinity to itself than to other clusters. We introduce a scalar parameter that allows us to have additional control on the rate of cluster merges. We consider a small fraction δ of $C^{AT}_{(\mathbf{s})jt}$ as external group evidence for j. The higher the value of δ, the stronger has to be the evidence to cause an existing author label to merge with another label or to split into two.

12 Experimental Evaluation

We begin by evaluating our algorithm on two real citation datasets. We compare our collaborative entity resolution model (**LDA-ER**) with the best attribute-based models. Next, to gain further understanding of the conditions under which entity resolution benefits from collaborative group information, we evaluate our model on a broad range of synthetic datasets with varying relational structure.

12.1 Results on Citation Data We first perform experimental evaluations on two citation datasets. The first is the CiteSeer dataset containing citations to papers from four different areas in machine learning, originally created by Giles et al. [15]. This has 2,892 references to 1,165 authors, contained in 1,504 documents. The second dataset is significantly larger; arXiv (HEP) contains papers from high energy physics used in KDD Cup 2003[2]. This has 58,515 references to 9,200 authors, contained in 29,555 papers. The authors for both datasets have been hand-labeled.

To evaluate our algorithms, we measure the performance of our model for detecting duplicates in terms of precision, recall and $F1$ on pairwise duplicate decisions. It is practically infeasible to consider all pairs, particularly for HEP, so as others have done, we employ a 'blocking' approach to extract the potential duplicates. This approach retains $\sim 99\%$ of the true duplicates for both datasets.

We use a simple scheme for attribute priors, where common last names are set to be 10 times more likely than other last names, and the free attribute '$\star$' is 10 times more likely than common names. When sampling group labels given the entity assignments at each step, we iterate until the log-likelihood converges. Typically for the first few steps, we perform 50 group sampling iterations for each author iteration. Thereafter we proceed with 20 group iterations for each author iteration. The $F1$ converges in about 30 author iterations for CiteSeer and 50 author iterations for HEP. On a 3.2GHz Dell Precision 670 Intel Xeon server, this takes between 2.5 and 10 minutes for CiteSeer and between 2 and 12 hours for HEP depending on the number of groups. As discussed in Section 11.2, we use a small fraction ($\delta = 0.5\%$) of group evidence for self probabilities.

As a baseline (**ATTR**), we compare with the hybrid *SoftTF-IDF* measure [8] that has been shown to outperform other unsupervised approaches for text-based entity resolution. Essentially, it augments the TF-IDF similarity for matching token sets with approximate token matching using a secondary string similarity measure. Jaro-Winkler is reported to be the best secondary similarity measure for *SoftTF-IDF*. We also experiment with the Jaro and the Scaled Levenstein measures. However, directly using an off-the-shelf string similarity measure for matching names results in very poor recall. From domain knowledge about names, we know that first and middle names may be initialed or dropped. A black-box string similarity measure would unfairly penalize such cases. To deal with this, **ATTR** uses string similarity only for last names and *retained* first and middle names. In addition, it uses drop probabilities p_{DropF} and p_{DropM} for dropped first and middle names, initial probabilities p_{FI} and p_{MI} for correct initials and p_{FIr} and p_{MIr} for incorrect initials. The probabilities we used are $0.75, 0.001$ and 0.001 for correctly initialing, incorrectly initialing and dropping the first name, while the values for the middle name are $0.25, 0.7$ and 0.002. We calculated the probabilities from the labeled datasets and then hand-tuned them for performance. Our observation is that baseline resolution performance does not vary significantly as these values are varied over reasonable ranges.

ATTR only reports pairwise match decisions, which are often inconsistent globally. We also evaluate a second baseline **ATTR*** which takes a transitive closure over the pairwise decisions in **ATTR**. Both **ATTR** and **ATTR*** need a

[1] http://en.wikipedia.org/wiki/List_of_most_popular_family_names
[2] http://www.cs.cornell.edu/projects/kddcup/index.html

similarity threshold for deciding duplicates and determining the right threshold is a problem for these algorithms. One of the strengths of **LDA-ER** is that it does not require any similarity threshold. For comparison, we consider the best $F1$ that can be achieved by the baselines over all thresholds.

Table 1: Performance of ATTR and ATTR* in terms of $F1$ using various secondary similarity measures with SoftTF-IDF. The measures compared are Scaled Levenstein (SL), Jaro (JA), JaroWinkler (JW) and the generative similarity model used with LDA-ER (Gen).

	CiteSeer			
	SL	JA	JW	Gen
ATTR	0.980	**0.981**	0.980	0.982
ATTR*	0.989	**0.991**	0.990	0.990
	HEP			
	SL	JA	JW	Gen
ATTR	**0.976**	0.976	0.972	0.975
ATTR*	**0.971**	0.968	0.965	0.970

Table 1 records baseline performance with various string similarity measures coupled with SoftTF-IDF. Note that the best baseline performance is with Jaro as secondary string similarity for CiteSeer and Scaled Levenstein for HEP. It is also worth noting that a baseline without initial and drop probabilities scores below 0.5 $F1$ using Jaro and Jaro-Winkler for both datasets. It is higher with Scaled Levenstein (0.7) but still significantly below the augmented baseline. Transitive closure affects the baseline differently in the two datasets. While it adversely affects precision for HEP, it improves recall for CiteSeer.

Table 2 shows the best performance of each of the three algorithms for each dataset. Note that the recall includes blocking, so that the highest recall achievable is 0.993 for CiteSeer and 0.991 for HEP. LDA-ER outperforms both forms of the baseline for both datasets for all string similarity measures and the improvements are statistically significant. For CiteSeer, **LDA-ER** gets close to the highest possible recall with very high accuracy. This means that it is able to retrieve almost all duplicates correctly. Improvement over the baseline is greater for HEP in terms of $F1$. Also, **LDA-ER** reduces error rate over the baseline by 22% for CiteSeer (from 0.9% to 0.7%) and by 20% for HEP (from 2.4% to 1.9%). Also, HEP has more than $64,6000$ true duplicate pairs, so that a 1% improvement in $F1$ translates to more than $6,400$ correct pairs.

Looking more closely at the resolution decisions from CiteSeer, we were able to identify some interesting combination of decisions by **LDA-ER** that would be difficult or impossible for an attribute-only model. There are instances in the dataset where reference pairs are very similar but correspond to different author entities. Examples include *(liu j, lu j)* and *(chang c, chiang c)*. **LDA-ER** correctly predicts that these are not duplicates. At the same time, there are other pairs that are not any more similar in terms of attributes than the examples above and yet are duplicates. These are also correctly predicted by **LDA-ER** by leveraging common collaboration patterns. The following are examples: *(john m f, john m st)*, *(reisbech c, reisbeck c k)*, *(shortliffe e h, shortcliffe e h)*, *(tawaratumida s, tawaratsumida sukoya)*, *(elliott g, elliot g l)*, *(mahedevan s, mahadevan sridhar)*, *(livezey b, livezy b)*, *(brajinik g, brajnik g)*, *(kaelbing l p, kaelbling leslie pack)*, *(littmann michael l, littman m)*, *(sondergaard h, sndergaard h)* and *(dubnick cezary, dubnicki c)*. An example of a particularly pathological case is *(minton s, minton andrew b)*, which is the result of a parse error. The attribute-only baselines cannot make the right prediction for both these sets of examples simultaneously, whatever the decision threshold, since they consider names alone.

Table 2: Performance of LDA-ER, ATTR and ATTR* for CiteSeer and HEP datasets. The standard deviation of the $F1$ is 3×10^{-4} for CiteSeer and 1.7×10^{-4} for HEP.

	CiteSeer			HEP		
	P	R	F1	P	R	F1
ATTR	0.990	0.971	0.981	0.987	0.965	0.976
ATTR*	0.992	0.988	0.991	0.976	0.965	0.971
LDA-ER	0.997	0.988	**0.993**	0.991	0.971	**0.981**

We were also interested in exploring how the number of collaborative groups affects the performance of our entity resolution algorithm. Table 3 records the performance of the group model on the two datasets with varying number of groups. While we observe a general trend where precision improves and recall suffers with more groups, note that the $F1$ is largely stable over a range of groups.

Table 3: LDA-ER Performance over varying number of groups

Num.	CiteSeer			HEP		
Grps	P	R	F1	P	R	F1
100	0.995	0.991	0.993	0.986	0.972	0.979
200	**0.997**	**0.988**	**0.993**	0.988	0.972	0.980
300	0.998	0.980	0.989	0.990	0.971	0.980
400	0.999	0.980	0.989	0.990	0.970	0.980
500				**0.991**	**0.971**	**0.981**
600				0.991	0.969	0.980

12.2 Properties of Collaborative Graphs While the LDA-ER model shows improvement for both citation datasets, the improvement is much more significant for the HEP dataset. On investigating why our model shows a larger

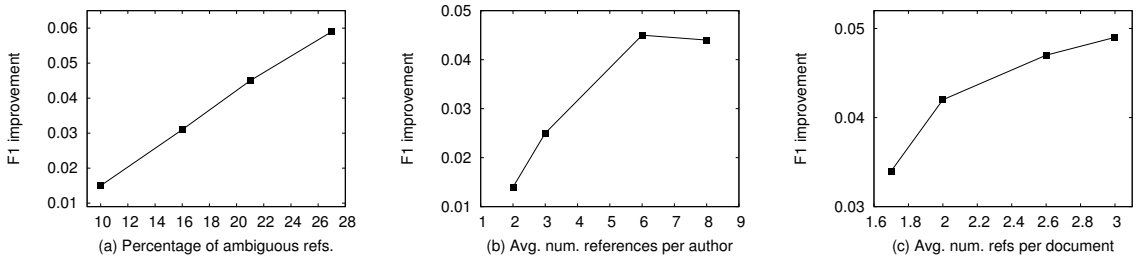

Figure 3: Improvement of LDA-ER over ATTR* for varying (a) ambiguity of references, (b) avg. number of references per author and (c) avg. number of references per document. Other parameters are held constant for each experiment.

improvement for HEP than for CiteSeer, we found some notable differences between the datasets. We call a reference ambiguous if there is more than one author entity with that last name and first initial. There is a significant difference in reference ambiguity between the two datasets — only 0.5% of the references in CiteSeer are ambiguous while 9% of HEP references are ambiguous. A second difference is in the density of the author collaboration graph. The average number of collaborators per author is 2.15 in CiteSeer and 4.5 in HEP. Finally, a third significant difference relates to the sample size. While the ratio of the number of references to the number of authors is 2.5 for CiteSeer, for HEP it is 6.36. On the other hand, one of the features that is preserved for both datasets is the average number of references per document, which is 1.9 for both.

In order to investigate which of these features is responsible for the performance difference, we ran our algorithm on a range of synthetically generated datasets. This allowed us to investigate the conditions under which our model is most likely to lead to significant improvements over algorithms which do not take into account collaborative structure. Due to space constraints, we provide only the outline of the data generator; it is reasonably sophisticated. It attempts to mimic the way authors of academic papers are generated by the underlying collaborative pattern among researchers. There are two phases in this generative process. First, a collaborative graph is created in steps, where in each step a collaborative edge is added between two authors. Each author is given a name sampled from US census data. By sampling from the top $k\%$ of this distribution we can control the percentage of ambiguous names in the data. Other parameters allow us to control the number of authors and the average collaboration degree. In the second stage, documents are created from this collaborative graph by first sampling an initiator author, who chooses randomly from collaborators to select co-authors for that document. The author names for each document are modified by a noise model to generate the references. Various parameters allow us to control the number of documents generated, the average number of authors per document and the level of noise in the references.

In our setup for experiments with synthetic data, we vary the synthetic dataset parameters one at a time holding the others constant. The default values of the parameters are set to reflect the features of the real datasets. The datasets have 1000 authors with an average of 4.5 collaborators. We generate 3000 documents with an average of 2 references per document and 15% ambiguous references. We explore varying the fraction of ambiguous references, the ratio of references to authors, the average number of collaborators and average number of references per document. Since the results are averaged over different datasets, we present only the improvement in $F1$ measure observed for the group model over ATTR*.

Figure 3 summarizes the trends that we observe. One significant improvement trend is over varying ambiguity in the references. As shown in Figure 3(a), it climbs sharply from 0.01 for 10% ambiguity (as in HEP) to 0.06 for 27% reference ambiguity. Figure 3(b) shows that LDA-ER naturally benefits from higher sample sizes for the author references. Figure 3(c) shows that LDA-ER benefits from a greater number of authors per document. However, no statistically significant trends emerged from our experiments with varying collaboration degree keeping other factors like sample size fixed; some experiments showed larger improvements with higher degree, however the results were not consistent. More thoroughly characterizing properties of the collaborative graph structure that lead to improved entity resolution is an interesting area for future work.

13 Conclusions

In this paper, we have developed a probabilistic generative model for collectively resolving entities in relational data. It is novel in that it does not make pair-wise decisions and introduces a group variable to capture relationships between entities. Our model may be viewed as extending the Dirichlet process mixture model to capture relations between entities or components. We propose an unsupervised approach for collective inference in our model that does not require any

labeled training data. In addition, we present a novel sampling strategy to estimate the number of entities automatically from the references. We have demonstrated the utility of the proposed model on two real-world citation datasets. Additionally, we have identified some of the conditions under which these models are expected to provide greater benefit. Areas for future work include extending the models to resolve multiple entity classes and better characterization of collaborative graphs amenable to these models.

References

[1] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *International Conference on Very Large Databases*, 2002.

[2] C. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2:1152–1174, 1974.

[3] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 2004.

[4] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *International Conference on Knowledge Discovery and Data Mining*, 2003.

[5] D. M. Blei and M. I. Jordan. Variational methods for the dirichlet process. In *International Conference on Machine Learning*, 2004.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:951–991, Jan 2003.

[7] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *International Conference on Management of Data*, 2003.

[8] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IJCAI Workshop on Information Integration on the Web*, 2003.

[9] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *International Conference on Knowledge Discovery and Data Mining*, 2002.

[10] A. Culotta and A. McCallum. A conditional model of deduplication for multi-type relational data. Technical Report IR-443, University of Massachusetts, 2005.

[11] A. Doan, Y. Lu, Y. Lee, and J. Han. Object matching for data integration: A profile-based approach. In *IJCAI Workshop on Information Integration on the Web*, 2003.

[12] X. Dong, A. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In *International Conference on Management of Data*, 2005.

[13] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

[14] T. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230, 1973.

[15] C. L. Giles, K. Bollacker, and S. Lawrence. CiteSeer: An automatic citation indexing system. In *Conference on Digital Libraries*, 1998.

[16] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, 2004.

[17] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In *International Conference on Management of Data*, 1995.

[18] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, 1999.

[19] H. D. III and D. Marcu. A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, 6:1551–1577, Sep 2005.

[20] D. V. Kalashnikov, S. Mehrotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *SIAM International Conference on Data Mining*, 2005.

[21] J. Kubica, A. Moore, J. Schneider, and Y. Yang. Stochastic link and group detection. In *National Conference on Artificial Intelligence*, 2002.

[22] X. Li, P. Morie, and D. Roth. Robust reading: Identification and tracing of ambiguous names. In *Human Language Technology Conference / NAACL*, 2004.

[23] A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *International Conference On Knowledge Discovery and Data Mining*, 2000.

[24] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems*, 2004.

[25] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Uncertainty in Artificial Intelligence*, 2001.

[26] A. E. Monge and C. P. Elkan. The field matching problem: Algorithms and applications. In *International Conference on Knowledge Discovery and Data Mining*, 1996.

[27] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9, 2000.

[28] Parag and P. Domingos. Multi-relational record linkage. In *ACM SIGKDD Workshop on Multi-Relational Data Mining*, 2004.

[29] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Neural Information Processing Systems*, 2003.

[30] P. Ravikumar and W. W. Cohen. A hierarchical graphical model for record linkage. In *Uncertainty in Artificial Intelligence*, 2004.

[31] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Uncertainty in Artificial Intelligence*, 2004.

[32] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *International Conference on Knowledge Discovery and Data Mining*, 2002.

[33] S. Tejada, C. A. Knoblock, and S. Minton. Learning object identification rules for information integration. *Information Systems Journal*, 26(8):635–656, 2001.

[34] W. E. Winkler. Methods for record linkage and Bayesian networks. Technical report, Statistical Research Division, U.S. Census Bureau, Washington, DC, 2002.