A Latent Dirichlet Model for Unsupervised Entity Resolution

Indrajit Bhattacharya
University of Maryland
College Park, MD, USA
indrajit@cs.umd.edu

Lise Getoor
University of Maryland
College Park, MD, USA
getoor@cs.umd.edu

ABSTRACT

In this paper, we address the problem of entity resolution, where given many references to underlying objects, the task is to predict which references correspond to the same object. We propose a probabilistic model for collective entity resolution. Our approach differs from other recently proposed entity resolution approaches in that it is a) unsupervised, b) generative and c) introduces a hidden 'group' variable to capture collections of entities which are commonly observed together. The entity resolution decisions are not considered on an independent pairwise basis, but instead decisions are made collectively. We focus on how the use of relational links among the references can be exploited. We show how we can use Gibbs Sampling to infer the collaboration groups and the entities jointly from the observed co-author relationships among entity references and how this improves entity resolution performance. We demonstrate the utility of our approach on two real-world bibliographic datasets. In addition, we present preliminary results on characterizing conditions under which collaborative information is useful.

1. INTRODUCTION

In many applications, there are a variety of ways of referring to the same underlying object. Given a collection of objects, we would like to a) determine the collection of 'true' underlying entities and b) correctly map the object 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 when 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 would like to determine when two records are referring to the same underlying individual (deduplication).

There is a long history of work in each of these research areas. Recently, general probabilistic approaches have been proposed [15, 22] as well as discriminative approaches [17, 21]. However, not all of these approaches explicitly capture links and collaborative information.

We introduce a generative probabilistic model for entity resolution. Our model builds on the recently proposed Latent Dirichlet Allocation model (LDA) [4]. While the LDA model was proposed for modeling documents as mixtures of topics, we adapt the model to the entity resolution problem. We motivate our approach on the task of resolving author references (which are the observed author names occurring in documents or document citations) in citation databases, but our model and algorithms are applicable in more general settings where noisy references to 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.

Our approach differs from existing approaches in that we explicitly leverage the underlying structure in the group interactions to improve the entity resolution performance. The group structure is learned from the observed collaborative relationships among entity references. For the case of author resolution, this means we make use of co-author relations to infer collaborative groups.

One contribution of our approach is that we propose an unsupervised collective entity resolution algorithm. 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. We also present a novel sampling algorithm for inferring the entities. Furthermore, unlike the majority of other approaches to entity resolution, the collaborative group model that we propose does not introduce a separate random variable for each pairwise resolution decision, but uses latent entity and group labels associated with each reference. We do not assume that equivalent strings necessarily refer to the same entity. In addition, part of the output of our algorithm is the set of entities and their canonical descriptions.

2. MOTIVATING EXAMPLE

In this section, we look at a concrete example of collective resolution of author references. Figure 1 shows four papers, each with its own author references. For instance, Paper P1 has three author references "Alfred Aho", "Jeffrey Ullman" and "S C Johnson". In all, we have ten author references that correspond to the three author entities: "Alfred V. Aho", "Jeffrey D. Ullman" and "S C Johnson". For example, all three references "A V Aho", "Alfred V Aho" and "Alfred Aho" correspond to the author entity named "Alfred V Aho". If we look at pairs of references individually and try to decide if they are duplicates, that may not be a difficult task

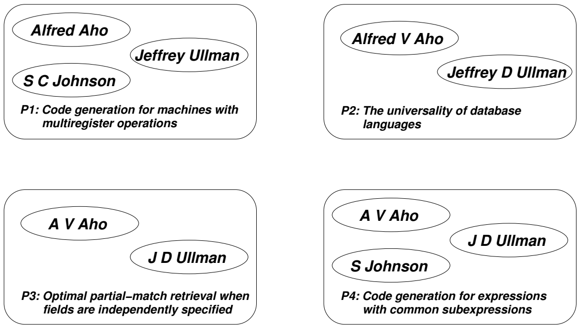

Alfred Aho
Jeffrey Ullman
S C Johnson
P1: Code generation for machines with multiregister operations
Alfred V Aho
Jeffrey D Ullman
P2: The universality of database languages
A V Aho
J D Ullman
P3: Optimal partial–match retrieval when fields are independently specified
A V Aho
J D Ullman
S Johnson
P4: Code generation for expressions with common subexpressions

Figure 1: An example author/paper resolution problem. Each box represents a paper reference (in this case unique) and each oval represents an author reference.

for uncommon names like "A V Aho" and "Alfred V Aho". But for frequently occurring names like "S Johnson" and "S C Johnson", it is a problem. While they will be duplicates in some cases, in others they will be distinct. We can however make use of additional evidence if we make these decisions *collectively*. Considering all the references together, we may decide that all the "Aho"s and the "Ullman"s correspond to the same entity, and *therefore* the two "Johnson"s are very likely to be references to the same author since they collaborate with the same set of author entities. This is what we would like to capture with our model. We would like to infer that the two "Johnson"s belong to the same collaborative *group* involving "Aho" and "Ullman" and use this additional evidence to predict that they correspond to the same author entity.

3. RELATED WORK

There is a large body of work on deduplication, record linkage, and co-reference detection. Here we review some of the main work, but the review is not exhaustive; for a nice summary report, see [27].

The traditional approach to entity resolution considers similarity of textual attributes. There has been extensive work on approximate string matching algorithms [19, 6] and adaptive algorithms that learn string similarity measures [3, 7, 26]. Beyond applying standard machine learning techniques, other approaches use active learning [25]. In addition, data integration is an area of active research [12, 19, 16].

The groundwork for posing record linkage as a probabilistic classification problem was done by Fellegi and Sunter[9]. Winkler[28] builds upon this work by introducing a latent match variable estimated using Expectation Maximization. More recently, hierarchical graphical models have been proposed [23].

Approaches that take relational features into account for data integration have been proposed [8, 5, 1, 20, 2]. Chaudhuri et al. [5] 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 [2].

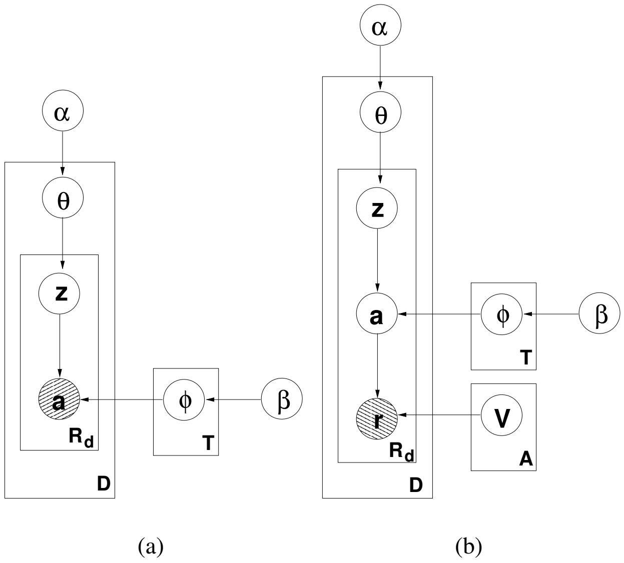
Probabilistic models that take into account interaction between different entity resolution decisions have been proposed for named entity recognition in natural language processing and for citation matching. McCallum et al.[17] employ conditional random fields (CRF) for noun coreference and use clique templates with tied parameters where the decision for one pair affects another through their overlap. Parag et al.[21] extend the CRF model to merge evidence across multiple fields. They are able to achieve significant benefit from generalizing the mapping of attribute matches to multiple references, for example being able to generalize from one match of the venue "Proc. of SIGMOD" with "Proceedings of the International Conference on Management of Data" to other instances.

Pasula et al.[22] 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. Li et al.[15] 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.

We model collaborative groups using LDA [4] which improves Probabilistic Latent Semantic Indexing [13] as a generative topic model for documents. The related author-topic model [24] notes the problem of duplicate authors; here we propose a solution for it. Kubica et al.[14] have proposed generative models for links using underlying groups, but they do not handle identity uncertainty.

4. LDA FOR AUTHORS

In this section adapt the LDA model for topics and words in documents to a group mixture model for author entities. We start with the simple case where there is no ambiguity in the author references. In the next section, we will expand the model to handle ambiguous author references and propose inference algorithms suited to the new model.



(a)

(b)

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

Consider a collection of D documents and a set of A authors corresponding to the authors of the documents. We have a set of R author references, $\{a_1, \ldots, a_R\}$. Each document can have multiple authors and for now, we assume the authors of each document are observed. For an author reference a_i, we use d_i to denote 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 given 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 the database being chosen for the j^{th} group is ϕ^j_i. We have T different multinomials, one for each group. Each paper d is modeled as a mixture over the 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^d_j. Each θ^d is drawn from a Dirichlet distribution with hyperparameters α; similarly each ϕ^j is drawn from a Dirichlet distribution with hyperparameters β.

5. LDA FOR AUTHOR RESOLUTION

So far, 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: 'Jonathan Elysia Smith', 'John E. Smith', 'J. Smith', etc. There may be mistakes due to typos or extraction errors. Finally, two 'J. Smith's may not refer to the same author entity. One may refer to 'John Smith' and another may refer to 'Jessica Smith'. 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 corrupts 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 corrupting the attribute v_a according to a noise model $\mathcal{N}$. We use a 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. This is represented in Figure 2(b).

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

$$\begin{aligned} P(\mathbf{r}; \alpha, \beta, \mathbf{V}) &= \prod_d P(\mathbf{r}_d; \alpha, \beta, \mathbf{V}) \\ &= \prod_d \sum_{\mathbf{a}_d} P(\mathbf{r}_d \mid \mathbf{a}_d; \mathbf{V}) P(\mathbf{a}_d; \alpha, \beta) \\ &= \int_\phi P(\phi; \beta) \prod_d \sum_{\mathbf{a}_d} P(\mathbf{r}_d \mid \mathbf{a}_d; \mathbf{V}) \\ &\times \int_\theta P(\theta; \alpha) P(\mathbf{a}_d \mid \theta, \phi) d\theta d\phi \qquad (1) \end{aligned}$$

6. INFERENCE USING GIBBS SAMPLING

In general, the integral in Eq. (1) is intractable due to coupling between θ and ϕ. Different approximations have been proposed, including variational methods [4], Gibbs Sampling [11] and Expectation Propagation [18].

We follow the approach proposed by [11] where θ and ϕ are not directly estimated as parameters. Instead, the posterior distribution $P(\mathbf{z}, \mathbf{a} \mid \mathbf{r})$ is first constructed and then θ and ϕ are estimated from this posterior distribution. Now, the joint probability can be derived from Eq. (1) as:

$$P(\mathbf{z}, \mathbf{a}, \mathbf{r}) = P(\mathbf{z}) P(\mathbf{a} \mid \mathbf{z}) P(\mathbf{r} \mid \mathbf{a}) \qquad (2)$$

where

$$P(\mathbf{z}) = (\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T})^D \prod_{d=1}^{D} \frac{\prod_t \Gamma(\alpha + C^{DT}_{dt})}{\Gamma(T\alpha + C^{DT}_{d*})} \qquad (3)$$

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

$$P(\mathbf{a} \mid \mathbf{z}) = (\frac{\Gamma(A\beta)}{\Gamma(\beta)^A})^T \prod_{t=1}^{T} \frac{\prod_a \Gamma(\beta + C^{AT}_{at})}{\Gamma(A\beta + C^{AT}_{*t})} \qquad (4)$$

is the conditional probability of the references 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 the references in document d and $C^{DT}_{d*} = \sum_t C^{DT}_{dt}$. 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 $\mathbf{a}_{-i}$ all but the i^{th} author label. In words, this is the probability that the i^{th} reference comes from the t^{th} group considering the current group and author assignment to *all other* references.

We can derive this full conditional distribution as

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

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 corrupted into the i^{th} reference.

Instead of sampling z_i and a_i as a block, they can be sampled separately:

$$P(z_i = t \mid \mathbf{z}_{-i}, \mathbf{a}, \mathbf{r}) \propto \frac{C^{DT}_{(-i)d_i t} + \alpha}{C^{DT}_{(-i)d_i *} + T\alpha} \frac{C^{AT}_{(-i)a_i t} + \beta}{C^{AT}_{(-i)*t} + A\beta} \qquad (5)$$

$$P(a_i = a \mid \mathbf{z}, \mathbf{a}_{-i}, \mathbf{r}) \propto \frac{C^{AT}_{(-i)at_i} + \beta}{C^{AT}_{(-i)*t_i} + A\beta} P(r_i \mid v_a) \qquad (6)$$

7. MODELING AUTHOR ATTRIBUTES

In the previous section, we assumed that the author attribute values v_a are known. But in general, the author attributes will not be known and will need to be *inferred* 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. (2), we derive the conditional distribution for assigning a value v to v_i given all author labels and references as:

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

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 "J.S. Smith" and "John Smith" references have been assigned author label i along with the reference "Jhon Smth", then the author attribute v_i is most likely to be "John S. Smith". 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, we set the attributes to a special value '$\star$'. We would like our model to prefer free authors over assigned authors and accordingly we assign a higher prior probability $P(\star)$ than all other attributes.

8. NOISE MODEL

The different ways for distorting or modifying an author attribute to a reference in a document is captured by the noise model $\mathcal{N}$. It 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}. These are expected to be lower than p_R. 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 mutating an author attribute v to a reference v' are obtained using edit-distance for strings. Then the mutation 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 will avoid formulating a separate elaborate procedure for searching over the number of authors and adapt it within our sampling framework.

9.1 Block Assignment for Entity Resolution

Instead of assigning labels to references individually, we will jointly (re)assign labels for a set of references. Specifically, we will pick 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 will assign new author labels to all references indexed by $\mathbf{s}$ simultaneously. Unfortunately, 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, we restrict the space of candidates such that that allow the set of references assigned to a particular author label may (a) merge with a set currently assigned to another author label, (b) stay unchanged or (c) split and have a portion assigned to a currently unassigned author label j'. In case (a), the number of authors is effectively decreased by one. In case (c), the number of authors is effectively increased by one. However, the number of possible partitions of $\mathbf{s}$ into j and j' is still $2^{|\mathbf{s}|}$. One simple but restricted solution is splitting to the set that last merged into label j via option (a). This is also the best partition in terms of the reference attributes.

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

$$T(t,i) = \prod_{n=1}^{C^{AT}_{(\mathbf{s})it}} (\beta + C^{AT}_{(-\mathbf{s})it} + C^{AT}_{(\mathbf{s})it} - n) \qquad (7)$$

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

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 $\mathbf{s}$. Then the conditional distribution can be derived as

$$P(\mathbf{a_s} = i \mid \mathbf{z}, \mathbf{a_{-s}}, \mathbf{r}) \propto \prod_{t \in \mathbf{z_s}} \frac{T(t,i)}{T(t,*)} \prod_{j \in s} P(r_j \mid v_i) \qquad (8)$$

9.2 An Interpretation of Block Assignment

The terms in this conditional probability can be rearranged so that the result makes intuitive sense. Let j be an index into $\mathbf{s}$ and t_j be the group label for that reference. Also, consider $\mathbf{s}$ to be an ordered set and denote by $\mathbf{s}_{<j}$ the set of elements in $\mathbf{s}$ strictly before position j. Then we may rewrite Eqn. 8 as

$$\begin{aligned} & P(\mathbf{a_s} = i \mid \mathbf{z}, \mathbf{a_{-s}}, \mathbf{r}) \\ \propto \; & \prod_{j \in s} \frac{\beta + C^{AT}_{(\mathbf{s}_{<j})it_j} + C^{AT}_{(-\mathbf{s})it_j}}{A\beta + C^{AT}_{(\mathbf{s}_{<j})*t_j} + C^{AT}_{(-\mathbf{s})*t_j}} P(r_j|v_i) \qquad (9) \end{aligned}$$

Here $C^{AT}_{(\mathbf{s}_{<j})it}$ is the number of times author label i and group label t have occurred jointly when looking at just the references in $\mathbf{s}_{<j}$. This may be interpreted as follows. We are assigning author labels to the references in $\mathbf{s}$ 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 added 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 Eqn. 9 reduces to Eqn. 6 when $\mathbf{s}$ has a single element.

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

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

In order to explicitly incorporate a preference for 'free authors' in our model, we observe that when one author label merges with another according to Eqn. 8, the attribute of the freed author j is set to '$\star$'. So the merge probability in Eqn 8 is augmented with an additional term: $P(\star)/P(v_j)$. Similarly, when splitting the references assigned to author j between j and currently unassigned j', we are taking away one free author. This is reflected by augmenting the split with the term $P(v_{j'})/P(\star)$. Observe that a higher prior probability of '$\star$' relative to other attributes favors merging and discourages splits.

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 hyperparameters. Their choice affects performance in different ways.

10.1 Number of Groups

Here we consider the effect of different numbers of groups. 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 Hyperparameters

To appreciate the roles of α and β, note from Eqn. 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 over fit the data. This is particularly undesirable for us, since we look to estimate the number of authors and need to generalize from the current author assignments. To get a feel for what are good values, 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 value of the parameters using domain knowledge, and then after each author sampling step, we re-estimate the parameter values looking at each reference and its author attribute. A weighted combination of the old parameter values and the newly estimated ones yields the parameter values for the next iteration.

11. ALGORITHM REFINEMENTS

Unlike the group labels, the author labels for references are sampled from a restricted space. Here we look at two ways to improve 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 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. In addition, bootstrapping allows us to estimate the maximum number of author labels that we want to use.

11.2 Group Evidence for Author Self Loops

In Eqn. 7, $C^{AT}_{(-\mathbf{s})at}$ is the number references outside $\mathbf{s}$ that have author label a and group label t. For any t, we may imagine this to be the group evidence for transition to author label a for $\mathbf{s}$. Consider the set $\mathbf{s}$ of references whose current author label is j and the term $T(t,j)$ for reassigning label j to them. Since $\mathbf{s}$ includes *all* references with author label j, $C^{AT}_{(-\mathbf{s})jt}$ will be 0 for all group labels t. Thus self-loops for author labels have a distinct disadvantage to other transitions in terms of group evidence. We may remedy this by considering 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 began by evaluating our algorithm on two citation datasets from different research areas. We compare our collaborative entity resolution model (**LDA-ER**) with models based solely on attributes. Next, to gain further understanding of the conditions under which entity resolution benefits from collaborative group information, we evaluated our algorithm on a broad range of synthetic datasets with varying relational structure.

[1] We use a list of common last names from http://en.wikipedia.org/wiki/List_of_most_popular_family_names

12.1 Results on Citation Data

We performed our first set of 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.[10]. 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.[3]

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 last names that occur in the common names list are set to be 10 times more likely than other names, and '$\star$' 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 75 author iterations for HEP. On a 1GHz Dell PowerEdge 2500 Pentium III server, this takes between 10 and 20 minutes for CiteSeer and between 8 and 20 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 [6] 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 arrived at these values by observing the true values in the datasets and then hand-tuning 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. Since the duplicate relation is transitive, we also evaluate **ATTR*** which removes inconsistencies in the pairwise match decisions in **ATTR** by taking a transitive closure. Note that this issue does not arise with **LDA-ER**; it does not make pairwise decisions. Both **ATTR** and **ATTR*** need a similarity threshold for deciding duplicates and determining the right threshold is a problem for these algorithms. We consider the best $F1$ that can be achieved 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 CiteSeer, **LDA-ER** gets close to the highest possible recall with very high accuracy. Improvement over the baseline is greater for HEP. While the improvement may not appear large in terms of F1, note that **LDA-ER** reduces error rate over the baseline by 22% for CiteSeer and by 25% for HEP. 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.

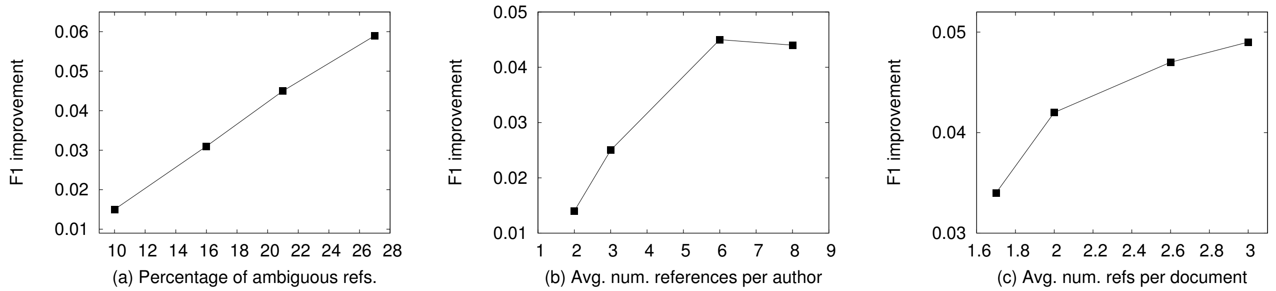
Table 2: Performance of LDA-ER, ATTR and ATTR* for CiteSeer and HEP datasets. The standard deviation of the F1 is 3×10^{-4} for CiteSeer and 1.7×10^{-4} for HEP.

	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**	0.992	0.972	**0.982**

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

[2] http://www.cs.cornell.edu/projects/kddcup/index.html

[3] We would like to thank Aron Culotta and Andrew McCallum for providing the author labels for the CiteSeer dataset and David Jensen for providing the author labels for the HEP dataset. We performed additional cleaning for both.



(a) Percentage of ambiguous refs.

(b) Avg. num. references per author

(c) Avg. num. refs per document

Figure 3: Improvement of LDA-ER over ATTR* in terms of F1 for varying (a) ambiguity of references, (b) average number of references per author and (c) average number of references per document. Other parameters are held constant for each experiment.

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.

We were also interested in exploring how the number of collaborative groups affect 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 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 dataset generator; it is reasonably sophisticated. [4] 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 corrupted 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 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*.

[4] We plan to make this generator available to other researchers

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. And 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. We believe that more thoroughly characterizing properties of the collaborative graph structure, which will lead to improved entity resolution, is an interesting area for future work.

13. CONCLUSIONS

In this paper, we have developed an unsupervised probabilistic generative model for entity resolution that is inspired by the LDA model. It is novel in that it exploits collaborative group structure for making resolution decisions. We have proposed a novel sampling algorithm for determining this group structure from observed collaboration relationships among ambiguous references. We have demonstrated the utility of the proposed model on two real-world citation datasets. 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.

14. REFERENCES

[1] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *VLDB*, 2002.

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

[3] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *KDD*, 2003.

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

[5] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *SIGMOD*, 2003.

[6] 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.

[7] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD*, 2002.

[8] 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.

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

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

[11] T. Griffiths and M. Steyvers. Finding scientific topics. In *Proc. of National Academy of Sciences*, 2004.

[12] M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In *SIGMOD*, 1995.

[13] T. Hofmann. Probabilistic latent semantic analysis. In *UAI*, 1999.

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

[15] X. Li, P. Morie, and D. Roth. Robust reading: Identification and tracing of ambiguous names. In *HLT-NAACL*, 2004.

[16] A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD*, 2000.

[17] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *NIPS*, 2004.

[18] T. P. Minka. Expectation propagation for approximate bayesian inference. In *UAI*, 2001.

[19] A. E. Monge and C. P. Elkan. The field matching problem: Algorithms and applications. In *KDD*, 1996.

[20] J. Neville, M. Adler, and D. Jensen. Clustering relational data using attribute and link information. In *Proceedings of the Text Mining and Link Analysis Workshop, IJCAI*, 2003.

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

[22] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *NIPS*, 2003.

[23] P. Ravikumar and W. W. Cohen. A hierarchical graphical model for record linkage. In *UAI*, 2004.

[24] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, volume 21, 2004.

[25] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *KDD*, 2002.

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

[27] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, Washington, DC, 1999.

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