ACM Journal of Data and Information Quality (ACM JDIQ), 4(2), 2013

Adaptive Connection Strength Models for Relationship-based Entity Resolution[†]

RABIA NURAY-TURAN
DMITRI V. KALASHNIKOV
SHARAD MEHROTRA
University of California, Irvine

Entity Resolution (ER) is a data quality challenge that deals with ambiguous references in data and whose task is to identify all references that co-refer. Due to practical significance of the ER problem, many creative ER techniques have been proposed in the past, including those that analyze relationships that exist among entities in data. Such approaches view the database as an entity-relationship graph, where direct and indirect relationships correspond to paths in the graph. These techniques rely on measuring the *connection strength* among various nodes in the graph by using a connection strength (CS) model. While such approaches have demonstrated significant advantage over traditional ER techniques, currently they also have a significant limitation: the CS models that they use are intuition-based fixed models that tend to behave well in general, but are very generic and not tuned to a specific domain, leading to suboptimal result quality. Hence, in this paper we propose an approach that employs supervised learning to adapt the connection strength measure to the given domain using the available past/training data. The adaptive approach has several advantages: it increases both the quality and efficiency of ER and it also minimizes the domain analyst participation needed to tune the CS model to the given domain. The extensive empirical evaluation demonstrates that the proposed approach reaches up to 8% higher accuracy than the graph-based ER methods that use fixed and intuition-based CS models.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications – Entity Resolution; H.2.8 [**Database Management**]: Database Applications – Data Mining

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Entity Resolution, Lookup, Reference Disambiguation, Connection Strength, Graph-based disambiguation, Relationship Analysis

1. INTRODUCTION

Real world raw data often has various data quality issues that need to be addressed before the data can be analyzed and processed. Failure to address such issues can lead to incorrect results and wrong conclusions in decision making. One very common data quality challenge is that of *Entity Resolution* (ER). The problem is that real-world raw data often refers to entities using references that can be ambiguous, e.g. “J. Smith”. The task of ER is to find all references that co-refer.

There has been a significant body of research on ER and related topics, see [Elmagarmid et al. 2007] for a survey of ER techniques. To resolve references, most of the traditional ER approaches rely on using simple feature based similarities (FBS) as the only source of disambiguation information [Jin et al. 2003; Chaudhuri et al. 2003; McCallum and Wellner 2003]. However, as of recently, new creative ways of using other sources of information have been proposed. They include using relationships among entities [Bhattacharya and Getoor 2004b; Minkov et al. 2006], external knowledge-bases [Elmacioglu et al. 2007], domain constraints [Chaudhuri et al. 2007], and entity behavior patterns [Yakout et al. 2010].

Our work builds on relationship-based techniques that view the database as a graph of entities that are linked to each other via relationships. We will refer to this graph-based ER framework as `ENRG`. For each given reference, the `ENRG` approach first utilizes the attribute similarities to identify a set of candidate entities that the reference might refer to. Then it employs graph theoretic techniques to discover and analyze the relationships between the

[†]This work was supported in part by NSF grant CNS-1118114 and a gift from Google.

references and the set of candidate entities. The solution is based on the hypothesis that a strong connection between two references is indicative that these references co-refer.

It is shown in [Kalashnikov and Mehrotra 2006] that the graph based relationship analysis outperforms the standard techniques such as the ones using feature-based similarities. However, the approach in [Kalashnikov and Mehrotra 2006] uses an intuition based mathematical model to measure the *connection strength* (CS). Given any two nodes u and v in graph G, the connection strength $c(u, v)$ returns how strongly u and v are interconnected to each other in G. In the past, CS was selected as variations of the random walk model [Minkov et al. 2006; Kalashnikov and Mehrotra 2006]. The CS measure is the essential component of the ENRG approach and the quality of the disambiguation depends on how suitable the selected CS model is for the underlying data. Hence as any other similarity/importance measure, CS is data-driven and requires the participation of a domain analyst either to choose the right CS model, or to tune a flexible CS model to the given dataset. However, deciding which model could benefit the most for the given dataset is a time-consuming job which requires participation of an expert. Further, using purely intuitive and fixed mathematical model can lead to sub-optimal results.

To overcome such problems, in this paper, we instead propose an *adaptive* connection strength model which *learns* the connection strength from past data which leads to significant quality improvement over the past non-adaptive models. The proposed approach minimizes the domain analyst's effort by adapting the CS measure automatically to the given domain using past data instead of requiring the analyst to decide which model is more suitable for the given domain and how to tune its parameters.

In addition, we address another limitation of the current ENRG framework [Kalashnikov and Mehrotra 2006]. Thus far, ENRG has been successfully applied to *single-type* ER only. In real life applications, however, references of multiple different types can be ambiguous. For instance, in the publication domain not only the authors but also the venues can be ambiguous, requiring disambiguation of reference of at least two distinct types. In this paper, we further study the proposed adaptive CS model for the *multi-type* ER problem and show that the ENRG and its adaptive version are suitable to resolve different types of ambiguities simultaneously.

In summary, **the main contributions** of this paper are:

- An *adaptive* graph-based ER approach is proposed, which shows significant improvement over baseline algorithms and intuition-based CS models.
- As a small but important contribution, the adaptive version of the ENRG framework is then extended to solve the *multi-type* ER problem.
- The effectiveness of the proposed approach is extensively evaluated.

The rest of the paper is organized as follows. Section 2 covers preliminary material on relationship-based ER techniques needed to explain our learning approach. Our main contribution, adaptive CS models and the algorithms for training them, are then developed in Section 3. Next, the proposed approach is empirically evaluated in Section 4. Section 5 covers related work. Finally, Section 6 concludes the paper.

2. PRELIMINARIES

Since the paper builds on the ENRG relationship-based framework, in this section we first summarize its key points in order to explain the proposed approach, which will be covered in Section 3.

2.1. Notation

The notation that we will be using is summarized in Table I. Assume that dataset $\mathcal{D}$ contains a set of entities X. Each entity $x \in X$ itself consists of one or more attributes, and it might also contain several references $x.r_1, x.r_2, \ldots, x.r_n$ to other entities in X. We

Table I. Notation Table

Notation	Description
r	Reference
x_r	Entity in whose context r is made
S_r	Set of options for r: $S_r = \{y_{r1}, y_{r2}, \ldots, y_{rn_r}\}$
y_{rj}	j-th option for r
g_r	Ground truth for r
w_{rj}	To-be-found weight of $r \rightarrow y_{rj}$ edge
$c(x_r, y_{rj})$	Connection strength between x_r and y_{rj}
T_i	i-th path type
w_i	To-be-learned weight for T_i
v_i	Probability to follow u-v paths of type T_i
c_i	Number of u-v paths of type T_i

denote the entity in the context of which reference r is made as x_r. These references can be referring to the entities of different types. Let R be the set of all references. Each reference $r \in R$ is essentially a description and may itself contain one or more attributes. For each reference $r \in R$ the *option set* S_r of that reference is identified. It contains all the entities in X to which r might potentially refer to: $S_r = \{y_{r1}, y_{r2}, \ldots, y_{rn_r}\}$. For r its S_r is initially determined either by blocking, domain knowledge, ad hoc techniques, or by choosing all entities whose feature-based similarity exceed a certain threshold. The ground truth entity g_r to which r refers to is unknown to the algorithm beforehand. The *goal* of the ER is to select the right y_{rj} (i.e., to find $y_{rj} = g_r$) from S_r to which r really refers.

The above definition is known as the *lookup* instance of ER, where the set of all entities is known beforehand and the task is for each reference to find which entity it refers to. Another common instance of the problem is known as *grouping*, where the set of entities is unknown, and the task is to find which references co-refer. While this paper focuses on the lookup case, similar techniques apply to the grouping instance as well.

2.2. ENRG Overview

ENRG represents the dataset $\mathcal{D}$ as an entity-relationship graph $G = (V, E)$, where V corresponds to the set of entities and E to the set of relationships among the entities. V is composed of two different types of nodes: regular nodes and choice nodes. Regular nodes correspond to some entity $x \in X$, while the choice nodes correspond to the references whose option set S_r contains at least two entities. Choice node for reference r reflects the fact that the ground truth g_r can be one of $\{y_{r1}, y_{r2}, \ldots, y_{rn_r}\}$.

A choice node is connected to the option nodes (i.e., the regular nodes that are in the option set) via edges, whose weights are initially undetermined. These edges are called *option edges* and their weights are called *option-weights*. Since the option-edges are semantically mutually exclusive, the sum of option-weights is 1: $w_{r1} + w_{r2} + \cdots + w_{rn_r} = 1$. The choice node is connected to the reference with an edge of weight 1. The option weights are variables whose values will be computed by ENRG framework, whereas the weights of the other edges in G are constants.

We assume that the input graph for ENRG is created using feature-based similarity approaches. Choice nodes are created for the references that are not solved with the FBS approaches. Then ENRG exploits the relationships iteratively for further disambiguation and outputs the fully resolved graph G.

Figure 1 demonstrates an entity-relationship graph for a toy publication dataset. The dataset consists of six papers:

(1) $P1$ by John Smith of CMU and John Black of MIT
(2) $P2$ by Susan Grey of MIT and “J. Smith”
(3) $P3$ by Jane Smith of Intel
(4) $P4$ by John Smith of CMU and Joe Brown

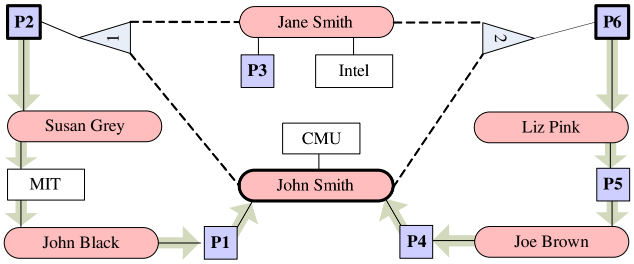

Fig. 1. An example graph for the publications domain.

```
ENRG($G, N_{iter}$)
    initialize $cs$ // set to 0
    for each reference $r$
        for each option $y_{rj} \in S_r$
            $P[x_r][y_{rj}] \leftarrow$ Find-All-Paths($G, x_r, y_{rj}, L$)

    for $i \leftarrow 0$ to $N_{iter}$ do
        for each reference $r$
            for each option $y_{rj} \in S_r$
                $\mathcal{P} \leftarrow P[x_r][y_{rj}]$
                $cs[x_r][y_{rj}] \leftarrow$ Get-CS($\mathcal{P}$)
            $E \leftarrow$ Determine-Equations($cs[x_r]$)
            $W \leftarrow$ Compute-Weights($E$)
    Resolve-References($G$)
```

Fig. 2. A simplified illustration of `ENRG` algorithm.

(5) $P5$ by Liz Pink and Joe Brown
(6) $P6$ by Liz Pink and "J. Smith"

The papers are represented as the square nodes in the graph. The authors are the rounded-rectangular nodes and their affiliations are the rectangular nodes. Two references two authors, "J. Smith" in $P2$ and "J. Smith" in $P6$ are uncertain. Thus, $P2$ and $P6$ are connected via triangular choice nodes "1" and "2" to the two possibilities: Jane Smith and John Smith.

`ENRG` framework builds on two important concepts: the *context attraction principle (CAP)* and *connection strength measure*, which are formally defined in [Kalashnikov and Mehrotra 2006] and which we summarize informally next. Given any two nodes u and v in graph G, the connection strength $c(u, v)$ returns how strongly u and v are interconnected to each other in G. The essence of CAP is that to decide which element $y_{rj} \in S_r$ is most likely to be the ground truth g_r for reference r, one can use the connection strength, e.g. by finding which $y_{rj} \in S_r$ maximizes $c(x_r, y_{rj})$.[1]

Figure 2 provides a simplified illustration of how the iterative version of `ENRG` approach works.[2] The algorithm first discovers the paths between each reference and its options and then uses them in the iterative weight computation procedure. Figure 2 illustrates that the relationship analysis is done in four steps:

(1) *Computing Connection Strength:* First, for each reference $r \in R$ and option $y_{rj} \in S_r$, the set P_L of all of the L-short simple paths[3] between x_r and y_{rj} is found. After that,

[1] In practice, the decision of which y_{rj} to pick as g_r can be based on multiple factors and not only on $c(x_r, y_{rj})$, but for simplicity of the subsequent discussion we will assume it is based only on $c(x_r, y_{rj})$.

[2] This algorithm is included to demonstrate the main relevant concepts, and it does not cover any of the many efficiency optimizations and other aspects.

[3] L-short simple paths are paths without loops of length not more than L.

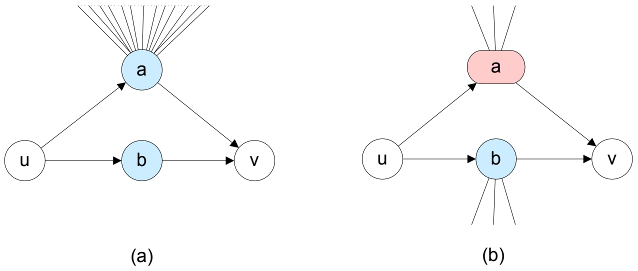

Fig. 3. Motivating Connection Strength Models. (a) Motivating WM (non-adaptive) model. (b) Motivating the need for adaptive CS models.

the connection strength $c(x_r, y_{rj})$ is computed using these paths. The result is a set of equations that relate $c(x_r, y_{rj})$ with the option weights.

(2) *Determining Equations for Options:* In the second step these equations and the CAP principle are used to determine a set of equations to relate the options to each other. ENRG uses the strategy where option weights are proportional to their CS values: $w_{rj} \cdot c(x_r, y_{rl}) = w_{rl} \cdot c(x_r, y_{rj})$.

(3) *Computing Option Weights:* In the third step these equations are solved and the values of $w_{rj} \in W$ are determined.

(4) *Resolve References:* Finally, the weights of the options are interpreted to resolve the references. That is, for each reference r the option $y_{rj} \in S_r$ with the maximum weight w_{rj} is selected as the entity r refers to.

2.3. Default Non-Adaptive Connection Strength Model

It is possible to use different models to compute the value of CS measure. In the previous study on which this paper is based, a fixed mathematical and intuition-based connection strength model has been used, which is called the *weight based model (WM)* or RandomWalk model. Let $P_L(u, v)$ be the set of all L-short simple u-v paths. WM model computes $c(u, v)$ as the probability to reach node v from node u via random walks in graph G. Accordingly, $c(u, v)$ is computed as the sum of the connection strength $c(p)$ of each path $p \in P_L(u, v)$, where $c(p)$ is the probability of following path p in G, that is:

$$c(u, v) = \sum_{p \in P_L(u,v)} c(p) \tag{1}$$

The probability of following a path p, given p exists, is computed as the product of probabilities of following an edge on p.

Though it is intuition-based, RandomWalk model is able to capture the importance of the paths quite well for most of the cases. For instance, consider the example in Figure 3(a) which illustrates a subgraph of G. It contains two paths between nodes u and $v : p_a = u \leftrightarrow a \leftrightarrow v$ and $p_b = u \leftrightarrow b \leftrightarrow v$. In this example, node a connects many nodes, not just u and v, whereas node b connects only u and v. Intuitively, we expect that the connection strength between u and v via b is more unique and thus stronger than the connection strength between u and v via a. RandomWalk model can easily capture that since it utilizes the degree of the intermediate nodes while computing the connection strength.

However, the RandomWalk model is not able to distinguish between the paths which are composed of different *types* of relationships if the degrees of the intermediate nodes in those paths are the same, as illustrated in Figure 3(b). Similarly, the figure contains two paths between nodes u and v. But now both nodes a and b are connected to the same number of nodes, so WM will not be able to distinguish them. However, the node *types*, as well as the types of relationships/edges in the two paths, are different. Assume that for this domain a

```
Get-CS($\mathcal{P}_L$)
static $W \leftarrow$ Load-Weights()
$c \leftarrow 0$
for each path $p \in \mathcal{P}_L$
    $i \leftarrow$ Get-Path-Type($p$)
    $c \leftarrow c + w_i$
return c
```

Fig. 4. Adaptive Get-CS for PTM.

```
Get-CS($\mathcal{P}_L$)
$c \leftarrow 0$
for each path $p \in \mathcal{P}_L$
    $v \leftarrow$ Get-Path-Prob($p$)
    $c \leftarrow c + v$
return c
```

Fig. 5. Get-CS for RandomWalk.

path going via a node of type b is more important than that of type a. But all of the current non-adaptive models (including WM), that do not analyze types of relationships and their significance for computing CS in a particular domain will fail to capture this knowledge. Consequently, in this paper we propose several flexible CS models that can capture this information and we develop supervised learning techniques for adapting these models to a given domain.

3. ADAPTIVE CONNECTION STRENGTH APPROACH

A connection strength model is the core component of ENRG framework. In this section we first discuss how to create a flexible highly-parameterized CS model that can be adapted to a given domain by choosing the right values for the parameters in Section 3.1. The parameterization is based on classifying paths into different types, which is explained in Section 3.2. Finally, we explain a linear programming-based supervised learning algorithm for adapting the model to the given domain by learning the values of the parameters in Section 3.3.

3.1. Adaptive Connection Strength Models

We observe that many of the non-adaptive CS models can be generalized, which can be used to create an adaptive connection strength model. Assume that we can classify each path that the disambiguation algorithm finds in graph G into a finite set of *path types* $S_T = \{T_1, T_2, ..., T_n\}$. If any two paths p_1 and p_2 are of the same type T_j, then they are treated as identical by the algorithm. Then, for any two nodes u and v, we can characterize the connections among them with a path-type count vector $Tuv = (c_1, c_2, ..., c_n)$, where each c_i is the number of paths of type T_i that exist between u and v. If we assume that there is a way to associate weight w_i with each path type T_i, then we can develop a basic parameterized CS model.

Basic Model. Basic adaptive model computes connection strength $c(u, v)$ as:

$$c(u,v) = \sum_{i=1}^{n} c_i w_i \qquad (2)$$

Figure 4 shows how this parameterized CS model, which we will refer to as *path type model* (PTM), can be implemented. Observe that we can make this model adaptive to a given domain by answering two questions:

(1) How to classify path into types and thus decide $S_T = \{T_1, T_2, ..., T_n\}$?
(2) How to choose weights $W = \{w_1, w_2, \ldots, w_n\}$ that work well for the given domain?

We will answer these questions in Section 3.2 and 3.3 respectively. Figure 4 illustrates that the main difference of the new model from the RandomWalk model shown in Figure 5 is that the importance of a path is not computed as the probability of following that path. Instead the connection strength of each path is learned on the past data and loaded into the weight vector W and utilized at the time of disambiguation.

```
Get-CS($\mathcal{P}$)
static $W \leftarrow$ Load-Weights()
static $\Gamma \leftarrow$ Load-Gammas()
$c \leftarrow 0$
for each path $p \in \mathcal{P}$
    $i \leftarrow$ Get-Path-Type($p$)
    $v \leftarrow$ Get-Path-Prob($p$)
    $c \leftarrow c + w_i + \gamma_i v$
return c
```

Fig. 6. Adaptive Get-CS for HM2.

The above formulation of the basic adaptive connection strength model, however, might not be ideal as it does take into account path types but now completely ignores the node degrees. But, as it has been illustrated in Figure 3(a), the paths going through edges connecting many nodes may not be as important as paths going through nodes with less connectivity. Thus, a more powerful adaptive CS model could potentially be developed by taking this information into account. Thus, we next develop two different hybrid adaptive connection strength models that combine the path type importance with the probability of following that path.

Hybrid Model 1 (HM1). Armed with the above intuition, this model mixes the PTM and RandomWalk models by computing $c(u, v)$ as:

$$c(u, v) = \sum_{i=1}^{n} v_i w_i \tag{3}$$

Here, v_i is the total probability to follow u-v paths of type T_i and w_i is the weight assigned to type T_i. So, for instance, if path of type T_i is completely unimportant, HM1 will assign $w_i = 0$ and eliminate this path from further consideration. Like RandomWalk, this model also takes into account the degree of nodes in the path by utilizes path probability in its v_i part.

Hybrid Model 2 (HM2). The second version of the hybrid model combines the RandomWalk with the basic adaptive CS model linearly:

$$c(u, v) = \sum_{i=1}^{n} c_i w_i + \gamma_i v_i \tag{4}$$

The intuition behind using a linear combination is that there is a tradeoff between using the node degree and path weight and this tradeoff can be best captured with the weighted sum of these different connection strength measures. Therefore, the model learns the importance w_i of each path type T_i as well as the importance of the WM part of that path by learning γ_i. This model has the flexibility of defaulting to different models including WM, PTM, and HM1 model. If, for a given domain, WM part is more important than the PTM part, then HM2 will automatically learn high values for γ_i's weights, and then HM2 will behave like WM. If the reverse is the case and PTM part is more important than WM, HM2 will learn γ_i's that are close to 0 and behave like PTM. Figure 6 shows the implementation of the HM2 model.

3.2. Path Classification

There are many possible ways to classify paths by utilizing the types of the edges and nodes present in those paths. For instance, the path classifier can view the paths as a set or sequence of edges and/or nodes. The most intuitive way to classify paths is representing the paths as a set of edges, by ignoring the order of these edges. We refer to this model

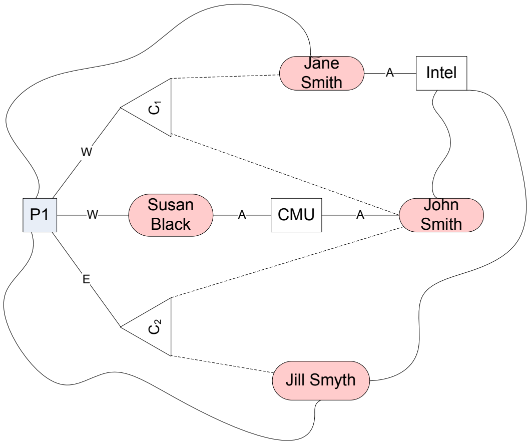

Fig. 7. Multi-type ER path type classification.

as the *edge type model* (ETM). This model has the flexibility of learning the importance of each edge type rather than the path type; so that the path importance is computed as the product of the edge importance weights. However, if the order of the edges plays a role in determining the path importance, which often is the case, ETM will not be able to capture that.

Accordingly, to be able to capture the order of edge types, this paper utilizes the *path type model* (PTM). PTM views a path as an ordered sequence of nodes and edges $\langle n_0, e_1, n_1, e_2, n_2 \ldots, e_k, n_k \rangle$. It then considers the node and edge types as a type string $\langle N_0, E_1, N_1, E_2, N_2, \ldots, E_k, N_k \rangle$, where N_i is the type of node n_i and E_i is the type of edge e_i. If two different paths have the same type string, they are considered to be of the same type, otherwise they are considered to be of different types. Each path type T_i is associated with its weight w_i. Computation of these weights is discussed in Section 3.

Example 1. Figure 7 demonstrates a subgraph in the publications domain. There, paper P_1 has two ambiguous references: one for the author (choice C_1) and the other one for the editor (choice C_2). The path between 'John Smith' and P_1 is composed of the following edge types: 'writes' (W) and 'affiliated with'(A). The node types are 'person'(Pe), 'paper' (P), and 'organization' (O). The whole path is then can be classified as of type that corresponds to the type string $\langle P, W, Pe, A, O, A, Pe \rangle$. □

Path Classification for Multi-type ER. Certain small adjustments need to be done in order to adapt the original single-type ENRG framework to handle multi-type ER cases. The problem with the original algorithm is that to disambiguate reference r it would consider connection strength $c(x_r, y_{rj})$ between x_r and y_{rj}. Let us illustrate the problem with that by considering what will happen to multi-type ER case illustrated in Figure 7 for choice nodes c_1 and c_2. For both of them, x_r is the node for paper P_1 and both of them have 'John Smith' as possible option. This means path P_1-Susan-CMU-John will be considered for disambiguating both of them. But this path has a certain type T_i which can have different importance when disambiguating author references compared to editor references and thus it should have at least two different weights w'_i and w''_i (one per each type of reference being disambiguated), and not one w_i. To avoid this problem for multi-type ER when disambiguating r we modify ENRG to consider $c(r, y_{rj})$ instead of $c(x_r, y_{rj})$. This effectively appends the type of reference being disambiguated to the beginning of $x_r \rightsquigarrow y_{rj}$ path type, and automatically solves the problem. For instance, for the case in Figure 7, instead of considering one $P_1 \rightsquigarrow$ John path

```
LEARN-PATH-WEIGHTS($G$)
  $I \leftarrow \emptyset$ // set of inequalities
  for each reference $r$
      for each option $y_{rj}$
          $\mathcal{P} \leftarrow$ FIND-ALL-PATHS($G, r, y_{rj}, L$)
          for $i \leftarrow 1$ to $n$ do
              $c_i \leftarrow 0$, $v_i \leftarrow 0$
          for each path $p \in \mathcal{P}$
              $i \leftarrow$ GET-PATH-TYPE($p$)
              $v_i \leftarrow v_i +$ GET-PATH-PROB($p$)
              $c_i \leftarrow c_i + 1$
      $I[r] \leftarrow$ CREATE-INEQUALITIES($C, V$)
  $\langle W, \Gamma \rangle \leftarrow$ SOLVE($I$)
  return $\langle W, \Gamma \rangle$
```

Fig. 8. Supervised learning algorithm for determining $W = \{w_i\}$ and $\Gamma = \{\gamma_i\}$ weights.

of type $\langle P, W, Pe, A, O, A, Pe \rangle$, the algorithm now will consider two paths of two different types: $\langle C_1, W, P, W, Pe, A, O, A, Pe \rangle$ and $\langle C_2, E, P, W, Pe, A, O, A, Pe \rangle$.

3.3. Learning Algorithm

Figure 8 illustrates the overall learning algorithm for HM2 model. It is applied to training data and learns the path weights $W = \{w_1, w_2, \ldots, w_n\}$ and $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ by reducing the problem to solving a linear programming problem. For that, it first classifies each found $r \rightsquigarrow y_{rj}$ path into its type by using PTM classification model. After that, for each $r \rightsquigarrow y_{rj}$ path of type T_i it computes the number of these paths c_i and the probability v_i of following those paths. Then these values are used to create the inequalities that are used in the linear programming problem. Finally, it uses a linear programming solver to solve the linear program that associates each path type with a weight.

The key question is how to create the corresponding inequalities to learn W and Γ. Since a supervised learning algorithm is employed, it works on a training/past data in which references are fully disambiguated. Hence, for each reference r in the training dataset the ground truth entity g_r it refers to and its possible options are known. The proposed approach uses the CAP principle to learn the CS model directly from data and applies it in the context of ER. Recall that the CAP principle states that for a reference r it is likely that

$$c(x_r, g_r) \geq c(x_r, y_{rj}) \text{ for all } r, y_{rj} \text{ s.t. } y_{rj} \in S_r \setminus \{g_r\}. \tag{5}$$

The employed entity resolution approach also states that for reference r the connection strength ‘evidence’ for the right option g_r should visibly outweigh that for the wrong ones $y_{rj} \in S_r \setminus \{g_r\}$. We can capture that by using the δ-band (“clear margin”) approach, by requiring that the difference between $c(r, g_r)$ and $c(r, y_{rj})$ should at least have a clear margin of δ:

$$c(r, g_r) - c(r, y_{rj}) \geq \delta \quad \text{for all } \ r, y_{rj} \text{ s.t. } y_{rj} \in S_r \setminus \{g_r\}. \tag{6}$$

However, in practice, because of the ‘likely’ part in the CAP, many of the inequalities in the System (6) should hold, but some of them might not. That is, System (6) might be overconstrained and might not have a solution. To address that, a slack is added to the inequalities and an additional constraint that requires the sum of the slacks should be minimized is introduced to the system, which then becomes:

$$
\begin{aligned}
&\text{Minimize } Z = \textstyle\sum_{rj} \xi_{rj},\\
&\text{subject to}\\
&c(r, g_r) - c(r, y_{rj}) + \xi_{rj} \geq \delta \text{ for all } \ r, y_{rj} \text{ s.t. } y_{rj} \in S_r \setminus \{g_r\}\\
&\text{and}\\
&\xi_{rj} \geq 0,\ 0 \leq w \leq 1.
\end{aligned}
\tag{7}
$$

System (7) always has a solution. Here, ξ_{rj} is a real-valued non-negative slack variable and the objective of the system is to find the W and Γ values such that the sum of these slack variables is minimized. System (7) essentially converts the learning task into solving the corresponding linear programming problem, and linear programming, in general, is known to have efficient solutions [Hillier and Lieberman 2001]. All $c(u, v)$ in System (7) should be computed according to Equations (2) (3) (4) and by adding a normalization constraint that all weights should be in $[0, 1]$ interval: $0 \leq w_i \leq 1$, for all i. The task becomes to compute the best combination of weights $w_1, w_2, \ldots, w_n$ that minimizes Z, which can be resolved by using any off-the-shelf linear programming solver.

3.4. The Shorter Path Importance

The experiments in [Kalashnikov and Mehrotra 2006] showed that the effect of longer paths to the disambiguation quality are marginal, hence the shorter connections are more influential than the longer paths. Accordingly, we now incorporate this intuition into the learning framework by adding an additional constraint to the linear programming problem. This constraint indicates that a path whose type is the prefix of a longer path type should be more important than the longer path. Then the overall linear programming model becomes:

$$
\begin{aligned}
&\text{Minimize } Z = \textstyle\sum_{rj} \xi_{rj},\\
&\text{subject to}\\
&c(r, g_r) - c(r, y_{rj}) + \xi_{rj} \geq \delta \text{ for all } \ r, y_{rj} \text{ s.t. } y_{rj} \in S_r \setminus \{g_r\}\\
&\text{and}\\
&\xi_{rj} \geq 0,\ 0 \leq w \leq 1\\
&w_i \geq w_j \text{ for all } i, j \text{ s.t. } w_i \text{ is a prefix of } w_j.
\end{aligned}
\tag{8}
$$

Example 2. To illustrate the above concepts, consider the small scenario demonstrated in Figure 9. It shows that x_1 has a reference r_1 that can potentially refer to y_1 or y_2 and, since it is training data, we know that it refers in reality to y_1. Similarly, it shows that reference r_2 is made in the context of x_2, and that r_2 matches y_3 and y_4, where in reality it is y_3. There are two different path types $T_1 = \langle e1, e3, e2\rangle$ and $T_2 = \langle e1, e2, e2\rangle$ in the graph.

For the first reference, there are two paths of same type T_1 connecting x_1 and y_1. Accordingly there is only one weight for both PTM part (w_1) and the WM part (γ_1). The probability of following the path p_1 is $\frac{1}{2} * \frac{1}{3}$, since one of the intermediate nodes is connected to 2 nodes, while the other one is connected to 3. Similarly the probability of the path p_2 is $\frac{1}{4}$. After combining these as explained in Equation (4), we will get the following equation: $c(x_1, y_1) = 2 * w_1 + \gamma_1 * (\frac{1}{6} + \frac{1}{4})$. Similarly for the x_1 and y_2 pair we have

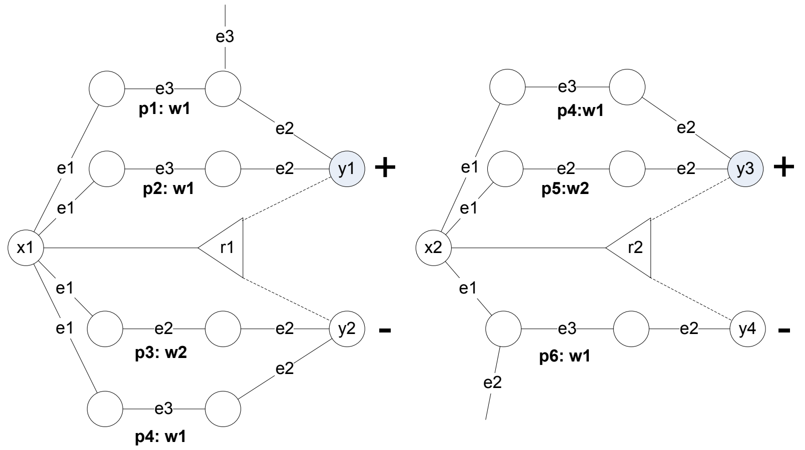

Fig. 9. A toy Example Graph Explaining the Linear Programming Approach for Path Weight Learning.

$c(x_1, y_2) = w_1 + \gamma_1 * \frac{1}{6} + w_2 + \gamma_2 * \frac{1}{4}$. The equations for x_2 are constructed in a similar fashion. We also know that the total weight of paths between x_1 (x_2) and y_1 (y_3) should outweigh that of between x_1 (x_2) and y_2 (y_4). Thus, we have:

$$\text{Minimize } Z = \xi_{11} + \xi_{21},$$

subject to

$$\begin{aligned} & w_1 - w_2 + \gamma_1 \tfrac{1}{4} - \gamma_2 \tfrac{1}{4} + \xi_{11} \geq \delta \\ & w_2 + \gamma_2 \tfrac{1}{4} + \xi_{21} \geq \delta \end{aligned} \tag{9}$$

and

$$0 \leq \xi_{11}, \xi_{21}, \; 0 \leq \gamma_1, \gamma_2, \quad \text{and} \quad 0 \leq w_1, w_2 \leq 1.$$

After solving Linear Program (9) for $\delta = 0.1$, we get $\gamma_1 = 0$, $\gamma_2 = 0$, $w_1 = 0.2$, and $w_2 = 0.1$ as one of the possible solutions. □

3.5. Improvement over the previous version of the approach

The model in [Nuray-Turan et al. 2007] has two different objectives and combines them linearly instead of using the δ-band approach, where the two objectives are: minimizing the sum of slack variables and maximizing the difference between $c(r, y_{rj})$ and $c(r, y_{rj})$, which translates to:

$$\text{Minimize } Z = \alpha \textstyle\sum_{rj} \xi_{rj} + (1 - \alpha) \sum_{rj} [c(r, y_{rj}) - c(r, g_r)],$$

subject to

$$c(r, g_r) - c(r, y_{rj}) + \xi_{rj} \geq 0 \text{ for all } r, y_{rj} \text{ s.t. } y_{rj} \in S_r \setminus \{g_r\} \tag{10}$$

and

$$\xi_{rj} \geq 0, \; 0 \leq w \leq 1.$$

Here α is a parameter that allows to vary the contribution of the two different objectives. It is a real number between 0 and 1, whose optimal value is learned by varying the α value on

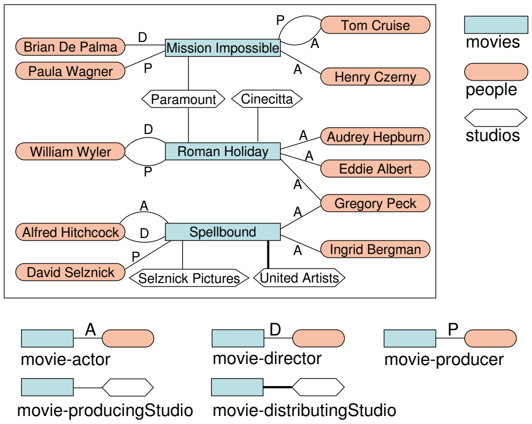

Fig. 10. Movies Dataset: Sample entity-relationship graph.

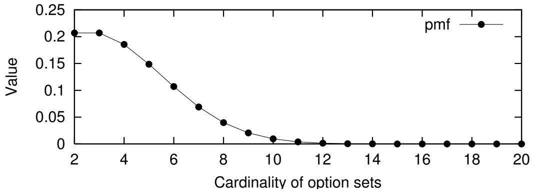

Fig. 11. PMF of sizes of option sets.

the training data and observing the effect on the quality of the disambiguation. Converting the linear programming δ-band approach, saves the algorithm the time for training for every different values of α, since the quality of the approach is largely independent of the selected δ. The reason is that δ is the only constant in System (8) and all the others (i.e., ξ, w, and γ) are variables. Increasing the δ value will scale these variables up, while decreasing the δ will scale them down.

4. EXPERIMENTS

We experimentally study our method using real datasets taken from two domains: Movies (Section 4.2.2) and Publications (Section 4.2.3). We first explain the experimental setup in Section 4.1 and then discuss the results in Section 4.2.

4.1. Experimental Setup

4.1.1. Datasets

— *MovData*. We use the Stanford Movies Dataset[2]. A sample entity-relationship graph for this dataset is illustrated in Figure 10. The dataset contains three different entity types: *movies* (11,453 entities), *studios* (992 entities) and *people* (22,121 entities) and there are five types of relationships: *actors, directors, producers, producing studios and distributing studios.*

— *PubData*. This real dataset is derived from two public-domain sources: CiteSeer [CiteSeer Dataset 2005] and HPsearch [HomePageSearch 2005]. It describes authors, their papers and their affiliations. It contains four different types of entities: *author* (14590 entities), *papers* (11682 entities), *departments* (3084 entities), and *organizations* (1494 entities) and

[2] http://www-db.stanford.edu/pub/movies/

four types of regular relationships: *author-paper*, *author-department*, *author-organization*, and *organization-department*.

4.1.2. Introducing Uncertainty. Both MovData and PubData datasets are clean datasets. When studying the quality of disambiguation, we use a method of testing commonly employed by many practitioners, including in a recent KDD Cup.[4] That is, we introduce uncertainty in the dataset manually in a controlled fashion and then analyze the resulting accuracy of various methods, as explained next.

We study the ER in different reference types such as *directors*, *distributing studios*, *producers*, and *authors*. For instance if we want to resolve references from movies to directors, we make the director entities uncertain. First, a fraction $f : 0 \leq f \leq 1$ of all director references is chosen to be made uncertain, while the rest remain certain. Each to-be-uncertain director reference r is made ambiguous by modifying it such that it either points to two directors instead of one (i.e., $c = |S_r| = 2$) or points to c directors where c is distributed according to the PMF in Figure 11 (i.e., $c = |S_r| \sim pmf$). Here c stands for the *cardinality* of S_r. Training and testing is performed for the same values of f and c, but the references chosen to be ambiguous are different in training and test data.

By using the above process, for each tested combination of f and c, we create 10 different training datasets out of (the entire) MovData dataset and 10 out of PubData. For each training dataset we create a test dataset. The reported results are the average over the 10 test datasets.

4.1.3. Baseline Systems. We compare the proposed connection strength models' performance with two different baselines. The first baseline is the default connection strength model, RandomWalk, used in [Kalashnikov and Mehrotra 2006]. The second baseline is an adapted version of the entity resolution algorithm proposed in [Bhattacharya and Getoor 2005]. The adapted algorithm works as follows. For each reference and an option, we compute the neighborhood similarity of the records they belong to by using DICE similarity. The neighborhood is defined as the common entities. For example if we are disambiguating the directors in movies dataset then the common entities can be distributing/producing studios, actors, producers and other directors because we compute the neighbor similarity of the movies that the reference and options belong to.

4.1.4. Evaluation Metrics. In this paper we focus primarily on the quality results that are reported in terms of *accuracy*[5], which is defined as the fraction of the correctly resolved ambiguous references. Efficiency is not the focus of this work. In terms of efficiency, the proposed HM2 model behave very similar to WM model, which has been studied extensively in [Kalashnikov and Mehrotra 2006].

It should be noted that, in general, it is expected that the quality improvements (demonstrated by ER techniques over baselines) will be large when the uncertainty in the datasets is large and the accuracy of the baseline models is low. However, when the accuracy of disambiguation is already high even for baselines, then it is expected that the quality improvement will be comparatively small. To test significance of the improvement over the previous approaches we use a standard statistical significance test, namely paired two-tailed t-test. We use the t-test to measure if the mean of two different evaluations are different. If the computed p-value is below some threshold (typically 0.10, 0.05, or 0.01) the difference is declared to be significant for that threshold value.

[4] KDD Cup is the annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD).

[5] Accuracy is the right measure for the lookup instance of ER. For the grouping instance, measures like pairwise F_1-measure and B-cubed are more appropriate.

Table II. Comparing the results of New Adaptive Model with the old models

Fraction	**Non-Adaptive (RandomWalk)**	**Old Adaptive**	**New Adaptive**
0.10	0.793	0.826	0.860 **(+0.024)**
0.25	0.778	0.828	0.858 **(+0.030)**
0.50	0.759	0.820	0.846 **(+0.026)**
0.75	0.720	0.795	0.826 **(+0.031)**
1.00	0.678	0.738	0.794 **(+0.056)**

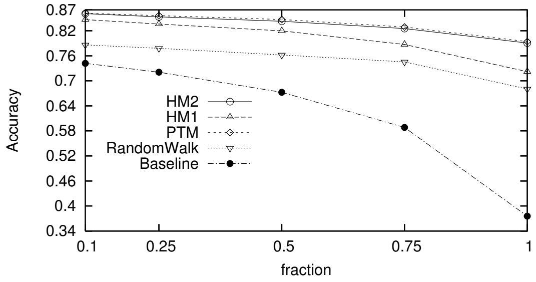

Fig. 12. MovData: Director References.

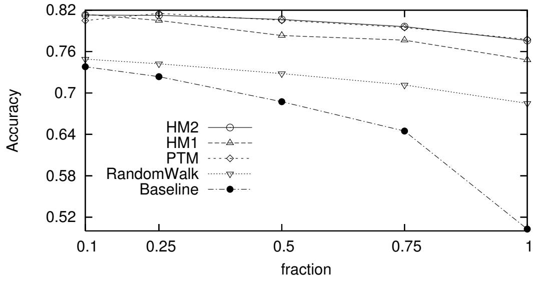

Fig. 13. MovData: Producer References.

4.2. Experimental Results

4.2.1. Improvement over the old model. Table II shows the improvement over the conference version of this paper for the director lookup problem. We refer to the model in [Nuray-Turan et al. 2007] as Old adaptive model. For the Old adaptive model we report the best accuracy result we have got by varying the α in $[0, 1]$ with the 0.001 intervals. In the table the results for the Non-adaptive CS measure (i.e., the RandomWalk) is also reported. The improvements over the previous column are shown in parenthesis. Overall improvement with the new model over the non-adaptive model is around 6.7-11.6%. These results show that the δ-band approach and the shorter path importance constraint added to the linear programming model improved the quality of the disambiguation compared to the old adaptive version.

4.2.2. Experiments on the Movies Dataset. In this domain, we study different lookup problems. We first compare the performance of the different algorithms on single type reference disambiguation such as directors, distributing studios and producers. Figures 12-14 show the quality results of these experiments.

In these experiments we selected $c \sim pmf$ as the cardinality of S_r and repeated the experiments for five different fractions of ambiguous entities ($f = \{0.1, 0.25, 0.5, 0.75, 1\}$). In the following we extensively study the results of each disambiguation task.

Figure 12 shows the quality results for director disambiguation in the movies domain.

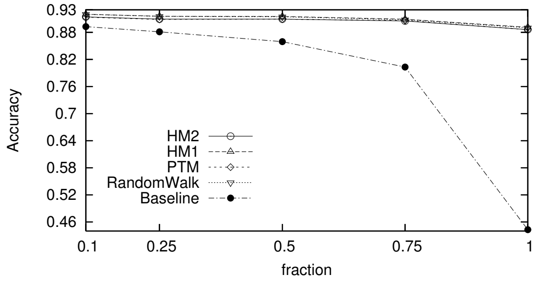

Fig. 14. MovData: Studio References.

Table III. Publications dataset results

Fraction	**HM2**	**HM1**	**PTM**	**RandomWalk**	**Baseline**
0.10	0.930	0.931	0.929	0.929	0.912
0.25	0.929	0.930	0.927	0.929	0.883
0.50	0.925	0.926	0.923	0.925	0.811
0.75	0.908	0.909	0.906	0.909	0.719
1.00	0.730	0.730	0.730	0.728	0.666

In these experiments HM2 and PTM models are the best and are indistinguishable; and they are followed by the HM1. All learning based connection strength models outperform the RandomWalk model and the Baseline. The mean difference of the learning models with respect to the RandomWalk and the Baseline models are statistically significant for $p \leq 0.01$, when measured with paired t-test. Since in this dataset PTM is better than the RandomWalk model, HM2 defaults to (behaves like) the PTM model. We observe similar behavior when we resolve the producer references (see Figure 13.) However, when we test the algorithms on the *studio* references, we observe that the mean difference of the adaptive CS approach with respect to the RandomWalk algorithm is negligible (see Figure 14). All four models perform equally well and significantly outperform the baseline algorithm. Another interesting observation is that the baseline algorithm is susceptible to the fraction of uncertain references. As the uncertain references increases the quality of the baseline algorithm drops drastically, whereas the connection strength based models' performance decreases gradually.

4.2.3. Experiments on the Publications Dataset. In this domain we executed experiments to disambiguate author references. The experiments were performed for different fractions of author uncertainty. We performed the experiments for $c = 2$ number of authors per choice node.

We repeated the experiments for 10 different test and training datasets. The average retrieval performance of different tests are reported in Table III. The results of learning models and RandomWalk model are essentially identical and are better than the baseline model. Since none of the learning based models is significantly better than the RandomWalk model, we can conclude that RandomWalk model is a good model for this specific setting. However, it may not work ideally for every instance of the publications domain. Observe that we see a similar behavior for the ER in studio references (see Figure 14).

4.2.4. The effect of Random Relationship on the Adaptive CS models. Our intuition is that when creating the PubData dataset, the analyst has chosen to project from CiteSeer relationships of only a few carefully chosen types that would work well with RandomWalk, while purposefully pruning away relationship types that are less important for disambiguation and would confuse RandomWalk model. In other words, the analyst has contributed his intelligence to

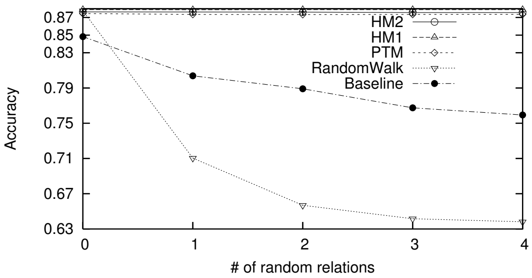

Fig. 15. Accuracy vs. Number of random relationships(noise).

that unintelligent model. To show that RandomWalk model is not ideal for every instances of the Publications domain, we have performed some additional experiments.

We gradually added random noise to the dataset, by introducing relationships of a new type – that represent random meaningless relationships. The random relationships were added to the 'false' cases only. That is, the added relationships are between the reference r and the candidates $y_{rj} \in S_r \setminus \{g_r\}$. We first added one random meaningless relationship to the 10% of the choice nodes, then added two random relationships, and so on. Figure 15 examines the effect of this noise on the effectiveness of RandomWalk as well as the other techniques. It shows the effect of random relationships up to 4 per choice node where the number of uncertain authors is defined with $c \sim pmf$. The figure shows that all of the learning models and the RandomWalk model obtain very high performance compared to the baseline approach. Initially, RandomWalk, HM1, HM2 and PTM models have the same accuracy. But as the level of noise increases, the accuracy of RandomWalk drops drastically below that of learned models as well as the baseline model. The learned models are based on an intelligent technique that learns the importance of various relationships and can easily handle noise – their curve stays virtually flat. Notice that baseline model's performance drops as well. That is because it also depends on the common neighbors of the reference and the object. Since there are more common neighbors between false options and the reference, as expected, the baseline model chooses the wrong options.

4.2.5. Why HM2 is the best option for the Adaptive CS. The previous experiments showed that either the PTM approach was significantly better than the RandomWalk or their difference was subtle. Accordingly, the HM2 model always resembled the PTM model. To show that the HM2 model always imitates the best performing model, we perform some additional experiments, which favor the WM model explicitly. These artificial datasets are created from the original ER problems. In these datasets, we explicitly make the edges typeless. That is, for instance, in movies dataset, there is no distinction between actors, producers, directors, and studios. The uncertain references are created as if the objective is still to resolve the selected type of reference (i.e. director, studio, etc.). That is the objective is to resolve the reference to the selected entity; however, their relevance with the entity is unknown. Figure 16 shows the results for 4 different ER tasks with only one relationship type (i.e., related-to).

Since RandomWalk and Baseline models are independent of the edge types their performances are same as before. For the director reference lookup, the HM1 and HM2 models are the best, whereas the PTM is the worst among all the `ENRG` based approaches. As expected HM2 model resembles the RandomWalk model and even perform better than that algorithm. For author and studio disambiguation, all 4 `ENRG` based models perform equally well, which means that for these cases even the simplest model, such as counting the paths between reference and the options, can get the same results which also explains the same quality results of all CS based models in the previous experiments. Finally, for the producer

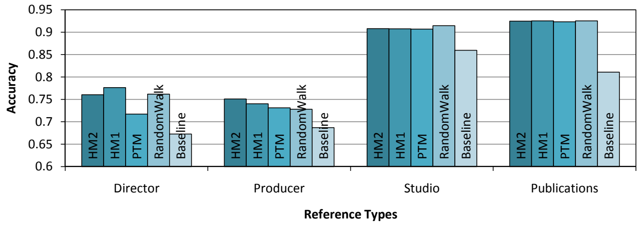

Fig. 16. Comparing HM2 with the other models in the absence of edge types (uncertain fraction =0.5).

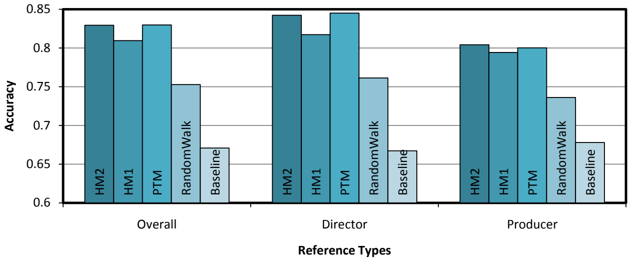

Fig. 17. Multi-type ER for director and producer references where $f = 0.5$.

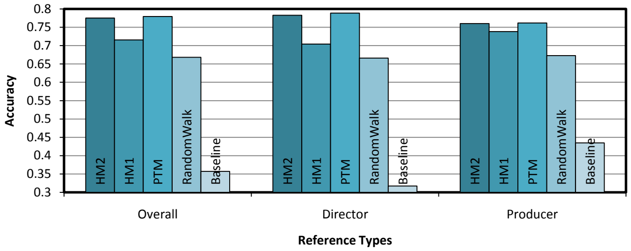

Fig. 18. Multi-type ER for director and producer references where $f = 1$.

lookup problem, although PTM and RandomWalk model perform equally well, HM1 and HM2 models are better.

4.2.6. Multi-type ER Experiments. In this section we study the effectiveness of the proposed approach on the Multi-type ER problem. In these experiments, we test the effectiveness the proposed approach when the references of different types are connected to the same type entity. Namely, we study the resolution of references to the *directors* and *producers*. Figures 17 and 18 show that PTM and HM2 are the best performing models and the improvement is statistically significant with respect to the other models, according to the t-test with $p < 0.01$. We observe a similar behavior when the approach is used to resolve the studio and director references simultaneously (see Figure 19).

5. RELATED WORK

Entity Resolution problem has been studied for the last 50 years under different names such as record linkage [Newcombe et al. 1959], duplicate detection [Bilenko and Mooney 2003], merge/purge problem [Hernandez and Stolfo 1995], hardening soft databases [Cohen et al. 2000], reference matching [McCallum et al. 2000], reference reconciliation [Dong et al. 2005], etc.

There has been a significant amount of work to improve the quality of the entity resolution techniques. Typically, the approaches use some textual similarity metrics to identify

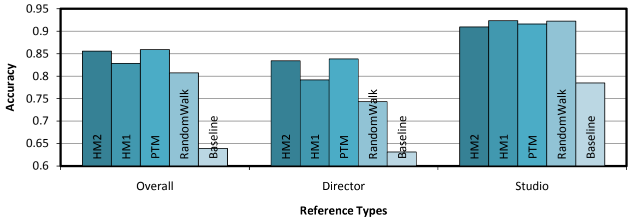

Fig. 19. Multi-type ER for director and studio references where $f = 0.5$.

the similar records [Chaudhuri et al. 2003; Ananthakrishna et al. 2002; Cohen et al. 2000]. These similarity metrics are identified for each field either manually by tuning the parameters or automatically by using different classifiers for each attribute field. To estimate the similarity between strings, several well-known string similarity measures have been used. These measures can be grouped into three different categories: (a) Character-based similarity metrics (such as edit distance [Levenshtein 1966] and its variants), (b) Token-based similarity metrics (such vector-space cosine similarity measures [Cohen et al. 2000], adaptive q-grams [Chaudhuri et al. 2003; Gravano et al. 2001]), and (c) Phonetic similarity metrics (such as Soundex [Russell and Odell]). Mostly these similarity metrics were used with fixed cost; however, [Cohen and Richman 2001] discusses how to combine different similarity metrics to identify duplicates and [Bilenko and Mooney 2003] proposes an adaptive tuning algorithm for field-similarity metrics. The probabilistic matching models such as [Newcombe et al. 1959] and [Fellegi and Sunter 1969] sees the duplicate detection as an Bayesian inference problem and Naive Bayes and Expectation Maximization are applied to the ER problem [Winkler 1993]. [Arasu et al. 2009] studies a greedy rule learning approach for string transformations such that these rules are used to convert a string to the other one. The approach is linear with the database size, thus scalable.

FBS approaches are suitable for the record linkage problems, where most of the fields of the records contain same fields; however, if the entities have ambiguous references to the entities in other tables, then it is most likely that the simple string similarity computations will not be sufficient to resolve these references. Thus, the researchers exploited new information sources to improve the confidence of two entities co-referring. Additional knowledge includes exploiting relationships between entities [Kalashnikov and Mehrotra 2006; On et al. 2007; Dong et al. 2005; Minkov et al. 2006], domain/integrity constraints [Shen et al. 2005; Fan et al. 2009], behaviors of entities [Yakout et al. 2010], and external knowledge bases such as ontologies and web search engines [Kanani et al. 2007; Elmacioglu et al. 2007]. Usually incorporating such new knowledge into the disambiguation process is costly, hence cleaning with additional knowledge is usually performed only on the entities that are still uncertain after FBS approaches are applied. For instance, exploiting web as external source is an expensive task since it requires querying the web search engines and it is a time consuming task due to network traffic [Elmacioglu et al. 2007; Kanani et al. 2007]. Techniques that utilize the functional dependencies [Fan et al. 2009], domain constraints [Shen et al. 2005], and aggregate constraints [Chaudhuri et al. 2007] apply to the record linkage problem by reasoning about the semantics of the data. For instance, aggregate constraints are used to make a merge/ do not merge decision of two records by comparing two different aggregation values associated with that pair. In [Yakout et al. 2010] record linkage problem is solved by finding the repeated patterns in different transaction logs; where the patterns completing each other are identified as co-referring.

Approaches that utilize relationships between entities for entity resolution view the database as an instantiated entity-relationship graph. The relationship analysis between entities is performed to compute the similarities of entities. Different versions of this model are discussed in [Malin 2005; Minkov et al. 2006; Bhattacharya and Getoor 2004a; Kalashnikov and Mehrotra 2006]. For instance, [Malin 2005] uses only one type of relationship (co-occurrence) with the shortest path analysis between the entities, whereas [Bhattacharya and Getoor 2004a] captures the neighbor similarity of entities using "writes" relationship in the publications domain. [Minkov et al. 2006] uses a lazy random walk on the graphs for "reference disambiguation" with the graphs, this model can be seen as an extension of the work in [Kalashnikov and Mehrotra 2006]. In [Herschel and Naumann 2008] the scalability of the graph based duplicate detection is studied. The studies in [Nuray-Turan et al. 2007; Yin et al. 2007] extend the relationship analysis proposed in [Kalashnikov and Mehrotra 2006]. A supervised learning algorithm for grouping task is proposed in [Yin et al. 2007], which uses the SVM with a linear kernel to learn the importance of each path whose importance is measured with the random walk probabilities in the feature vector. That is [Yin et al. 2007] uses the HM1 model with SVM for the grouping problem. The authors did not discuss the path classification model they utilized.

In this paper, we employ the approach presented in [Kalashnikov and Mehrotra 2006] to test our adaptive connection strength model. The algorithm uses a graphical methodology; the disambiguation decisions are made not only based on object features like in the traditional approach, but also based on the inter-object relationships, including indirect ones that exist among objects. The essence of the adaptive model is to be able to learn the importance of various connections on past data in the context of reference disambiguation. This paper is an extended version of the paper in [Nuray-Turan et al. 2007] and our novel contributions are:

— Linear Programming model is changed. Now it has only one objective function and uses the δ-band approach, which decreased the training effort drastically.
— The shorter path importance constraint is added to the linear program.
— A new adaptive connection strength model, HM2, is proposed.
— The adaptive CS approach is extended for multi-type ER.
— Thorough analysis of the proposed approach is performed with extensive experiments.

Some of our past entity resolution and web people search work is also related, but not directly applicable and uses different methodologies [Kalashnikov and Mehrotra 011 ; Nuray-Turan et al. 2011; Nuray-Turan et al. 2012; Kalashnikov et al. 2008; Chen et al. 2007; Kalashnikov et al. 2007; Chen et al. 2005; 2009; Nuray-Turan et al. 2009; Kalashnikov et al. 2009].

6. DISCUSSIONS AND CONCLUSION

In this paper we have developed an adaptive version of the ENRG framework. Our results show that adaptive connection strength model always outperforms the state-of-the-art RandomWalk model and the baseline approach. We also demonstrate that the HM2 model always defaults to the best performing model and, further, the HM1 and HM2 models are more robust compared to the RandomWalk and PTM models. There are many advantages of self-tunable CS model in the context of entity resolution. First of all, it minimizes the analyst participation, which is important since nowadays various data-integration solutions are incorporated in real Database Management Systems (DBMS), such as Microsoft SQL Server DBMS [Chaudhuri et al. 2005]. Having a less analyst-dependent technique makes that operation of wide applicability, so that non-expert users can apply it to their datasets. The second advantage of such a CS model is that it expects to increase the quality of the disambiguation technique. There are also less obvious advantages. For example, the technique is able to detect which path types are marginal in their importance. Thus, the algorithm that discovers paths when computing $c(u, v)$ can be sped up, since the path search space can be reduced by searching only for important paths. Speeding up the algorithm that discovers paths is important since it is the bottleneck of the overall disambiguation approach [Kalashnikov et al. 2005; Kalashnikov and Mehrotra 2006].

REFERENCES

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

Arasu, A., Chaudhuri, S., and Kaushik, R. 2009. Learning string transformations from examples. In *VLDB*.

Bhattacharya, I. and Getoor, L. 2004a. Deduplication and group detection using links. In *10th ACM SIGKDD Workshop on Link Analysis and Group Detection (LinkKDD-04)*.

Bhattacharya, I. and Getoor, L. 2004b. Iterative record linkage for cleaning and integration. In *DMKD Workshop*.

Bhattacharya, I. and Getoor, L. 2005. Relational clustering for multi-type entity resolution. In *MRDM Workshop*.

Bilenko, M. and Mooney, R. 2003. Adaptive duplicate detection using learnable string similarity measures. In *SIGKDD*.

Chaudhuri, S., Ganjam, K., Ganti, V., Kapoor, R., Narasayya, V., and Vassilakis, T. 2005. Data cleaning in Microsoft SQL Server 2005. In *SIGMOD*.

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

Chaudhuri, S., Sarma, A. D., Ganti, V., and Kaushik, R. 2007. Leveraging aggregate constraints for deduplication. In *SIGMOD Conference*.

Chen, S., Kalashnikov, D. V., and Mehrotra, S. 2007. Adaptive graphical approach to entity resolution. In *Proc. of ACM IEEE Joint Conference on Digital Libraries (JCDL 2007)*. Vancouver, British Columbia, Canada.

Chen, Z., Kalashnikov, D. V., and Mehrotra, S. 2005. Exploiting relationships for object consolidation. In *Proc. of International ACM SIGMOD Workshop on Information Quality in Information Systems (ACM IQIS 2005)*. Baltimore, MD, USA.

Chen, Z. S., Kalashnikov, D. V., and Mehrotra, S. 2009. Exploiting context analysis for combining multiple entity resolution systems. In *Proc. of ACM SIGMOD International Conference on Management of Data (ACM SIGMOD 2009)*. Providence, RI, USA.

CiteSeer Dataset 2005. CiteSeer Dataset. `http://citeseer.ist.psu.edu/oai.html`.

Cohen, W., Kautz, H., and McAllester, D. 2000. Hardening soft information sources. In *SIGKDD*.

Cohen, W. and Richman, J. 2001. Learning to match and cluster entity names. In *ACM SIGIR-2001 Workshop on Mathematical/Formal Methods in Information Retrieval*.

Dong, X., Halevy, A. Y., and Madhavan, J. 2005. Reference reconciliation in complex information spaces. In *SIGMOD*.

Elmacioglu, E., Kan, M.-Y., Lee, D., and Zhang, Y. 2007. Web based linkage. In *WIDM07*.

Elmagarmid, A., Ipeirotis, P., and Verykios, V. 2007. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering*, 1–16.

Fan, W., Jia, X., Lo, J., and Ma, S. 2009. Reasoning about record matching rules. In *VLDB*.

Fellegi, I. and Sunter, A. 1969. A theory for record linkage. *Journal of Amer. Statistical Association 64,* 328, 1183–1210.

Gravano, L., Ipeirotis, P., Jagadish, H., Koudas, N., Muthukrishnan, S., and Srivastava, D. 2001. Approximate string joins in a database (almost) for free. In *Proceedings of the international conference on very large data bases*. 491–500.

Hernandez, M. and Stolfo, S. 1995. The merge/purge problem for large databases. In *SIGMOD*.

Herschel, M. and Naumann, F. 2008. Scaling up duplicate detection in graph data. In *CIKM*. 1325–1326.

Hillier, F. and Lieberman, G. 2001. *Introduction to operations research*. McGraw-Hill.

HomePageSearch 2005. `http://hpsearch.uni-trier.de`.

Jin, L., Li, C., and Mehrotra, S. 2003. Efficient record linkage in large data sets. In *DASFAA*.

Kalashnikov, D. V., Chen, Z., Mehrotra, S., and Nuray, R. 2008. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE) 20,* 11.

Kalashnikov, D. V., Chen, Z., Nuray-Turan, R., Mehrotra, S., and Zhang, Z. 2009. WEST: Modern technologies for Web People Search. In *Proc. of the 25th IEEE Int'l Conference on Data Engineering (IEEE ICDE 2009)*. Shanghai, China. demo publication.

Kalashnikov, D. V. and Mehrotra, S. 2006. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (ACM TODS) 31,* 2, 716–767.

KALASHNIKOV, D. V. AND MEHROTRA, S. 2011-. Sherlock @ UCI: data cleaning and entity resolution project at uc irvine. http://sherlock.ics.uci.edu.

KALASHNIKOV, D. V., MEHROTRA, S., CHEN, S., NURAY, R., AND ASHISH, N. 2007. Disambiguation algorithm for people search on the web. In *Proc. of the IEEE 23rd International Conference on Data Engineering (IEEE ICDE 2007)*. Istanbul, Turkey. short publication.

KALASHNIKOV, D. V., MEHROTRA, S., AND CHEN, Z. 2005. Exploiting relationships for domain-independent data cleaning. In *SIAM International Conference on Data Mining (SIAM Data Mining 2005)*. Newport Beach, CA, USA.

KANANI, P., MCCALLUM, A., AND PAL, C. 2007. Improving author coreference by resource-bounded information gathering from the web. In *IJCAI*.

LEVENSHTEIN, V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*. Vol. 10. 707–710.

MALIN, B. 2005. Unsupervised name disambiguation via social network similarity. In *Workshop on Link Analysis, Counterterrorism, and Security*.

MCCALLUM, A. AND WELLNER, B. 2003. Object consolidation by graph partitioning with a conditionally-trained distance metric. In *KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation*.

MCCALLUM, A. K., NIGAM, K., AND UNGAR, L. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *ACM SIGKDD*.

MINKOV, E., COHEN, W. W., AND NG, A. 2006. Contextual search and name disambiguation in email using graphs. In *SIGIR*.

NEWCOMBE, H. B., KENNEDY, J. M., AXFORD, S. J., AND JAMES, A. P. 1959. Automatic linkage of vital records. *Science 130*, 954–959.

NURAY-TURAN, R., CHEN, Z., KALASHNIKOV, D. V., AND MEHROTRA, S. 2009. Exploiting Web querying for Web People Search in WePS2. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

NURAY-TURAN, R., KALASHNIKOV, D. V., AND MEHROTRA, S. 2007. Self-tuning in graph-based reference disambiguation. In *Proc. of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007), Springer LNCS*. Bangkok, Thailand.

NURAY-TURAN, R., KALASHNIKOV, D. V., AND MEHROTRA, S. 2012. Exploiting web querying for web people search. *ACM Transactions on Database Systems (ACM TODS)*. accepted.

NURAY-TURAN, R., KALASHNIKOV, D. V., MEHROTRA, S., AND YU, Y. 2011. Attribute and object selection queries on objects with probabilistic attributes. *ACM Transactions on Database Systems (ACM TODS) 36,* 4.

ON, B.-W., KOUDAS, N., LEE, D., AND SRIVASTAVA, D. 2007. Group linkage. In *ICDE*.

RUSSELL, R. AND ODELL, M. Soundex. *US Patent 1*.

SHEN, W., LI, X., AND DOAN, A. 2005. Constraint-based entity matching. In *AAAI*.

WINKLER, W. 1993. *Matching and record linkage*.

YAKOUT, M., ELMAGARMID, A. K., ELMELEGY, H., OUZZANI, M., AND QI, A. 2010. Behavior based record linkage. In *VLDB*.

YIN, X., HAN, J., AND YU, P. 2007. Object distinction: Distinguishing objects with identical names. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 1242–1246.