A Generic Web-Based Entity Resolution Framework

Denilson Alves Pereira
Department of Computer Science, Federal University of Lavras, Lavras, Brazil.
E-mail: denilsonpereira@dcc.ufla.br

Berthier Ribeiro-Neto
Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil and Google Engineering, Belo Horizonte, Brazil. E-mail: berthier@dcc.ufmg.br

Nivio Ziviani, Alberto H.F. Laender, and Marcos André Gonçalves
Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil.
E-mail: {nivio, laender, mgoncalv}@dcc.ufmg.br

Web data repositories usually contain references to thousands of real-world entities from multiple sources. It is not uncommon that multiple entities share the same label (polysemes) and that distinct label variations are associated with the same entity (synonyms), which frequently leads to ambiguous interpretations. Further, spelling variants, acronyms, abbreviated forms, and misspellings compound to worsen the problem. Solving this problem requires identifying which labels correspond to the same real-world entity, a process known as entity resolution. One approach to solve the entity resolution problem is to associate an authority identifier and a list of variant forms with each entity—a data structure known as an *authority file*. In this work, we propose a generic framework for implementing a method for generating authority files. Our method uses information from the Web to improve the quality of the authority file and, because of that, is referred to as WER—Web-based Entity Resolution. Our contribution here is threefold: (a) we discuss how to implement the WER framework, which is flexible and easy to adapt to new domains; (b) we run extended experimentation with our WER framework to show that it outperforms selected baselines; and (c) we compare the results of a specialized solution for author name resolution with those produced by the generic WER framework, and show that the WER results remain competitive.

Introduction

In large-scale data repositories, it is often hard to recognize distinct references to the same real-world entity. For instance, online product catalogs are expected to generate information related to products of interest to the user. However, given that similar labels can be used to refer to distinct products in different Web pages, products that are not related to the user intention might appear in the answer set. To illustrate, while "HP Officejet J3680 All-in-One Printer, Fax, Scanner, Copier," "HP CB071A#A2L J3680 Officejet Multifunction Printer," and "Hewlett Packard Officejet 3680 All-in-One Printer (CB071A)" refer to the same printer, "HP Officejet J4580 All-in-One Printer, Fax, Scanner, Copier" refers to a different one.

Digital libraries, which need to keep bibliographic citation metadata collected from several sources, face a similar challenge. Besides replicated records, it is usual to find ambiguous author names in bibliographic citations. Ambiguity may occur due to the existence of multiple authors with the same name (polysemes) or different name variations for the same author (synonyms)—a problem referred to in the literature as the "entity resolution problem" (Benjelloun et al., 2009; Bhattacharya & Getoor, 2007; Bilenko, Mooney, Cohen, Ravikumar, & Fienberg, 2003; Tejada, Knoblock, & Minton, 2001).

To further illustrate, consider the set of bibliographic references of a book composed of several chapters written by different authors. Chapter authors use distinctive variant forms, or labels, to refer to the various conferences and journals containing the articles they used. We would like to normalize the labels to avoid confusion and misinterpretation. An appealing solution is to associate with each conference or journal a unique identifier as well as a list of the variant labels used to refer to it—a data structure known as an "authority file" (Auld, 1982; French, Powell, & Schulman,

Received December 03, 2009; revised November 20, 2010; accepted January 12, 2011

 DOI: 10.1002/asi.21518

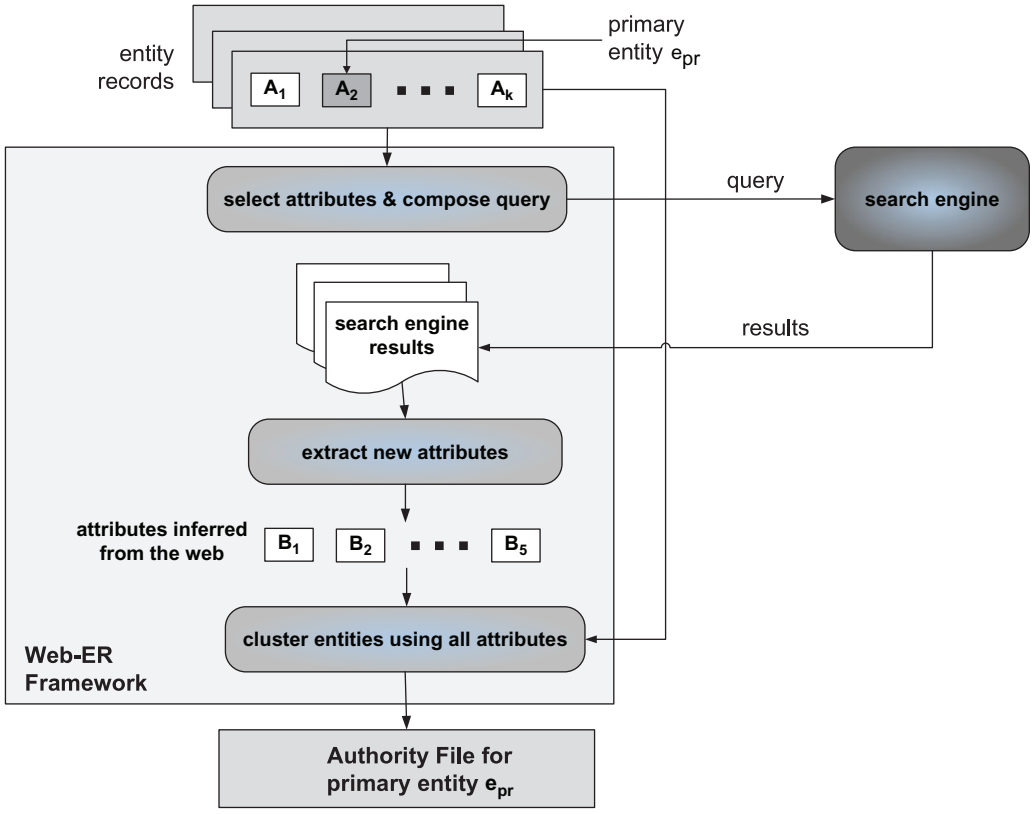

FIG. 1. The WER Framework, which combines the domain-specific attributes A_i with the Web-inferred attributes B_j to produce authority files of higher quality.

2000). The original bibliographic references are the input to the problem. In our approach, they are used to generate Web search queries, whose results contain information that we use to compose a higher quality authority file.

Characterization of the Problem

Given a set of entity references, such as product labels or conference titles, entity resolution is the process of identifying which of them correspond to the same real-world entity. Owing to the existence of relations among distinct entity references, they might appear grouped in an entity record. To illustrate, a bibliographic citation record is normally composed of names of authors, article title, and publication venue name, which are all distinct types of entities of the world. Their grouping into a citation record is necessary to provide all the information related to the corresponding publication event.

Whenever entity records are composed of references to distinct types of entities, it is necessary to indicate which entity type we intend to disambiguate, i.e., the entity type that is the target of the entity resolution solution. In this work, we refer to it as the *primary entity reference* e_{pr}.

In case of replicas, some researchers also deal with the problem of merging the replicated entity references to generate a canonical form for each entity (Benjelloun et al., 2009; Wick, Culotta, Rohanimanesh, & McCallum, 2009). In this article, however, we aim at generating authority files without considering the merging of replicated entity references.

Our Solution

In this article, we discuss the design, implementation, and validation of a generic and configurable framework for solving the entity resolution problem, which we refer to as WER—Web-based Entity Resolution. The WER framework, illustrated in Figure 1, works as follows.

Input: A set of entity records, each record composed of domain-specific attributes A_i, and a primary entity reference e_{pr} to disambiguate (indicated as A_2 in Figure 1).

1. for each record do: (a) select one or more attributes and use them to compose a query, (b) submit the query to a search engine, and (c) collect the top m documents in the answer set;
2. parse the documents in the answer set and extract (Web-inferred) attributes B_j such as URLs, titles, texts of the documents, names, and acronyms used to refer to the entities;
3. using the original attributes A_i and the Web-inferred attributes B_j, cluster the entity records relative to the primary entity e_{pr}, such that each cluster corresponds to a single real-world entity;
4. in each cluster, select an attribute value to be the canonical entity name, and output the respective entry to the authority file.

DOI: 10.1002/asi

The approach is singular because it combines the original domain-specific attributes with the attributes extracted from the Web, which allows us to improve the clustering procedure and, thus the quality of the authority file. In fact, experimental results show that the WER solution outperforms selected baselines on three distinct datasets: printer descriptions, publication venue titles, and author names.

The WER framework is built in modules to allow the easy and convenient development of solutions to new domains. That is, the framework can be quickly adapted to take on new entities distinct from the ones we tested here. Because of this characteristic, we say that it is *generic*.

An alternative is to build a code that only works for a specific entity type. The code in this case is more specific and can be fine tuned to take advantages of particularities inherent to the entity type in case. However, its application to new entity types (in new domains) is not immediate because more extensive adaptation of the code is required. We refer to this solution as a *specific* solution. Our previous work in Pereira, Ribeiro-Neto, Ziviani, and Laender (2008) and in Pereira et al. (2009) was based on specific solutions for distinct entity types. In this article, we reimplement those solutions using the generic WER framework to show that the results are still very positive. Further, in the specific case of author name resolution, we directly compare the results produced by specific and generic implementations of our WER approach.

The WER Framework

In the following subsections we discuss our solution to the entity resolution problem within the context of the WER framework.

The WER Formulation

In the entity resolution problem, we have a set of entity records $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$, in which each entity record r_i has attributes $A_1, A_2, \ldots, A_k$, which we refer to as domain-specific attributes. To refer to the jth attribute of record r_i, we adopt the notation $r_i A_j$. To illustrate, in the case of bibliographic citation records, the domain-specific attributes are author name, work title, publication venue title. As each domain-specific attribute might be a reference to a distinct entity type, it is necessary to indicate the entity type to be disambiguated, which we refer to as the *primary entity reference* e_{pr}. That is, e_{pr} is also provided as an input.

The *entity resolution problem* consists of

1. determining the set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ of distinct real-world entities related to the primary entity reference e_{pr}, and
2. associating with each entity record r_i the corresponding (correct) entity $e_j \in \mathcal{E}$.

Notice that the number m of distinct entities is not provided as an input.

An authority file (Auld, 1982; French et al., 2000) is a set of authority records, in which each record representing an entity is composed of a heading (to be used as the authority label), and a list of variant labels used to refer to the entity, called cross references. The headings and the cross references can be used by a search system to answer queries on the entity.

Solving the entity resolution problem requires clustering records to identify cross references to the same entity. In the previous approaches in the literature, this clustering procedure is guided by the domain-specific attributes A_i that compose the entity records. In the WER approach, on the other hand, the clustering procedure is guided not only by the A_i attributes, but also by additional attributes (related to the primary entity reference e_{pr}) obtained from the Web.

The operation of the WER framework is illustrated in Figure 1 and can be described in greater details, as follows.

- *Step 1* (Collecting information from the Web). Given an entity record r_i, we select a subset of its domain-specific attributes to compose a query, which is submitted to a search engine. The top m documents in the answer set are recorded. Notice that a separate query is produced for each entity record in the input.
- *Step 2* (Extracting information from the documents). The Web pages returned for each query are parsed with the purpose of extracting new attributes that can also be used to describe the primary entity. These attributes, which we refer to as $B_1, B_2, \ldots, B_5$ because they are always in number of 5, are inherent characteristics of the Web pages such as the URL, title, text, an acronym if it exists, and a heading for the entity reference. The heading, which is obtained from the text of the Web page, is computed as the string in the text that is most similar to the primary entity reference in the record. To compute the similarity between two strings, we opt for the Jaccard similarity coefficient (Jaccard, 1901). The $r_i.B_1$ to $r_i.B_5$ attributes are generically referred to as Web-inferred attributes.
- *Step 3* (Clustering data). Using the values of the domain-specific and Web-inferred attributes, a clustering procedure is applied to group entity records such that each cluster represents a distinct real-world entity $e_j \in \mathcal{E}$. The clustering procedure works as follows:
  - compute a pairwise similarity $sim(r_i, r_j)$ between any pair of references $r_i, r_j \in \mathcal{R}$, and
  - use the $sim(r_i, r_j)$ similarities to execute a clustering procedure such as KNN.

  The similarity function $sim(r_i, r_j)$ is based on a linear combination of various ranking functions, each of which assigns a score to a pair of attribute values of r_i. Let $F_p(r_i, r_j)$ be one of these ranking functions, and let w_p be a weight associated with F_p. Then, we define:

$$\begin{cases} sim(r_i, r_j) = \sum_p w_p \times F_p(r_i, r_j) \\ \sum_p w_p = 1 \end{cases}$$

That is, we use a simple linear combination of ranking functions and determine the best weighted combination empirically. Our objective is to demonstrate that the Web-inferred attributes obtained from the Web are useful to

DOI: 10.1002/asi

disambiguate entities, independently of the strategy used to combine them;
- Step 4 (Generating an authority file). For creating an authority file, a heading is selected for each generated cluster, considering that a cluster represents an authority record. The heading of each authority record is selected among the headings of the entity references in the cluster as being the most common heading.

In the immediately following subsection, we discuss steps 2, 3, and 4 of the WER framework in greater detail.

Extracting Information From Web Pages in the Answer Set

From each document (Web page) $d_j \in \mathcal{D}$ (the set of all collected documents), we extract values for the following attributes: URL ($d_j.B_{url}$), title ($d_j.B_{title}$), text content ($d_j.B_{text}$), and heading ($d_j.B_{head}$). We might also consider an attribute that is specific to the input data. For instance, for the case of conferences and journals, an acronym is a specific attribute that is of great value for disambiguation purposes. This is referred to as acronym ($d_j.B_{acro}$). The extraction of the first three attributes is direct. Attribute $d_j.B_{url}$ is the URL associated with the document, attribute $d_j.B_{title}$ is the title of the document, i.e., the text between the HTML tags <title> and </title>, and attribute $d_j.B_{text}$ is the text of the document. The other two attributes are extracted as follows.

Heading extraction. To extract a heading associated with the document $d_j \in \mathcal{D}$ (the set of the top m documents in answer set of the query associated with r_i), we first break the text of the document into phrases, where each phrase is formed by a sequence of words delimited by punctuation marks. To illustrate, consider that d_j contains the following text:

> *"ACM/IEEE Joint Conference on Digital Libraries. ACM/IEEE 2003 Joint Conference on Digital Libraries (JCDL 2003), 27–31 May 2003, Houston, Texas, USA, Proceedings. IEEE Computer Society 2003, . . . "*

From this document, we extract phrases such as: *"ACM/IEEE Joint Conference on Digital Libraries",. . . , "Houston", "Texas", "USA", "Proceedings", and "IEEE Computer Society 2003"*.

After that, we compute the similarity between each phrase and the value of the attribute of r_i related to the primary entity reference e_{pr}. The similarity is based on the Jaccard similarity coefficient. Then, we select the phrase with the highest similarity to be the value of the heading attribute $d_j.B_{head}$. For example, if the attribute value related to e_{pr} were "Joint Conference on Digital Libraries," the extracted heading would be "ACM/IEEE Joint Conference on Digital Libraries." We also expand simple abbreviations, like those that match the beginning of a word. If the expansion succeeds, we will consider the expanded form as a candidate for heading.

When the value of the attribute of r_i related to e_{pr} is a short text (one or two words) and one of these words is identified as an acronym, instead of obtaining the most similar phrase as described below, we expand the acronym. The expansion algorithm matches each letter, or sequence of letters, in the acronym with the initial letters in the words of the phrases extracted from d_j. The phrase that matches more letters of the acronym, beyond a given threshold, is selected as the heading attribute $d_j.B_{head}$. For the well-known entity acronyms, the search engine usually returns the acronym and the long entity name in the documents because the two forms are often together. For example, if the attribute value related to e_{pr} were "JCDL," the extracted heading for the previous example would be "ACM/IEEE Joint Conference on Digital Libraries," since "JCDL" is an acronym and its letters match with the initial letters in "Joint Conference Digital Libraries."

Acronym extraction. From the text of a Web page d_j in the results set, we extract an acronym (if it exists) that matches the extracted heading. Candidates for acronym must have at least two uppercase letters. The expansion method matches initial letters, simple abbreviations, and common conversions such as "2" and "to." This is also the approach adopted by Larkey, Ogilvie, Price, and Tamilio (2000). An expansion coefficient is computed as the number of acronym symbols that were expanded. The acronym with the highest expansion coefficient, higher than a given threshold, is selected to represent the acronym associated with the document ($d_j.B_{acro}$).

Attribute values extracted from the documents are associated with the entity record r_i, as follows.

URLs, titles, and texts. The URLs, titles, and texts associated with the entity record r_i are obtained from the documents $d_j \in \mathcal{D}_i$, as:

$$r_i.B_{url} = \bigcup_j d_j.B_{url}$$

$$r_i.B_{title} = \bigcup_j d_j.B_{title}$$

$$r_i.B_{text} = \bigcup_j d_j.B_{text}$$

Heading. The heading associated with the entity reference r_i is selected as the most common value among the values of the attribute $d_j.B_{head}$. We consider the most common value as the heading with the highest sum of similarities with regard to other headings extracted from $\mathcal{D}_i$. Let $simH(d_j.B_{head}, d_k.B_{head})$ be a function that returns the similarity between the strings $d_j.B_{head}$ and $d_k.B_{head}$. Then, we compute the sum of similarities among the strings $d_j.B_{head}$ and $d_k.B_{head}$, $\forall d_j, d_k \in \mathcal{D}_i$, as follows:

$$sumSim(d_j.B_{head}) = \sum_{k \neq j} simH(d_j.B_{head}, d_k.B_{head}) \quad (1)$$

Next, the value of the "heading" attribute for the entity record r_i is computed as

$$r_i.B_{head} = d_j.B_{head}$$

such that $d_j.B_{head}$ has the maximum $sumSim(d_j.B_{head})$.

DOI: 10.1002/asi

Acronym. The acronym associated with the entity record r_i is selected as being the most common value among the values of the attributes $d_j.B_{acro}$, computed as follows. Let $aCount(d_j.B_{acro})$ be a function that counts the number of occurrences of each distinct and not a null acronym $d_j.B_{acro}$. Then, the value of the "acronym" attribute for the entity record r_i is computed as

$$r_i.B_{acro} = d_j.B_{acro} \tag{2}$$

such that $d_j.B_{acro}$ has the maximum $aCount(d_j.B_{acro})$.

Clustering Entity References

For clustering entity records (Step 3), we use the domain-specific and the Web-inferred attributes. Ideally, each cluster should represent a distinct real-world entity. The clustering procedure works by computing the pairwise similarity between entity records r_i and r_j. For this, a similarity function $sim(r_i,r_j)$ based on a linear combination of ranking functions is used. Each ranking function associates a score with attribute values of the same attribute. Let $F_p(r_i,r_j)$ be the ranking function for the pth attribute of entity records r_i and r_j, and let w_p be a weight associated with the pth attribute. Define:

$$\begin{cases} sim(r_i, r_j) = \sum_p w_p \times F_p(r_i, r_j) \\ \sum_p w_p = 1 \end{cases} \tag{3}$$

The ranking functions for the URL, title, and acronym attributes are as follows:

$$F_{url}(r_i, r_j) = \begin{cases} 1.0 & \text{if } |r_i.B_{url} \cap r_j.B_{url}| \geq q, \quad q > 0 \\ \frac{p}{q} & \text{if } |r_i.B_{url} \cap r_j.B_{url}| = p, \quad 0 \leq p < q \end{cases}$$

where q is a parameter set empirically, corresponding to the required number of common URLs in the two sets.

$$F_{title}(r_i, r_j) = simT\,(r_i.B_{title}, r_j.B_{title})$$

We define the functions $F_{text}(r_i, r_j)$ and $F_{head}(r_i, r_j)$ using similarity functions $simX(r_i.B_{text}, r_j.B_{text})$, $simH(r_i.B_{head}, r_j.B_{head})$, respectively, where *simT*, *simX*, and *simH* are any string similarity function such as the Jaccard coefficient. This is analogous to the definition of $F_{title}(r_i, r_j)$.

Finally,

$$F_{acro}(r_i, r_j) = \begin{cases} 1.0 & \text{if } r_i.B_{acro} = r_j.B_{acro} \\ & \text{and } r_i.B_{acro} \text{ is not null} \\ 0.0 & \text{if } r_i.B_{acro} \neq r_j.B_{acro} \\ 0.7 & \text{if } r_i.B_{acro} \text{ is null} \\ & \text{and } r_j.B_{acro} \text{ is null} \end{cases}$$

For clustering the entity records, we can adopt distinct clustering procedures. For instance, we could use either *K*-nearest-neighbor (KNN) clustering or hierarchical agglomerative clustering (HAC) (Croft, Metzler, & Strohman, 2009).

Generating an Authority File

In Step 4, we generate an authority file. Since each cluster c_k generated in Step 3 represents a distinct real-world entity $e_k \in \mathcal{E}$, it also represents an authority record ar_k of the authority file. Thus, we store a link to the entity records in that cluster. Also, we store the values of the primary entity references, which comprises the cross references attribute $ar_k A_{cross}$.

Analogous to Equations (1) and (2), we compute the heading attribute $ar_k A_{head}$ for the authority record ar_k, where each $r_i.B_{head}$ is the heading of the entity record $r_i \in c_k$, as follows:

$$sumSim(r_i.B_{head}) = \sum_{j \neq i} simH(r_i.B_{head}, r_j.B_{head})$$

$$ar_k.A_{head} = r_i.B_{head}$$

such that $r_i.B_{head}$ has the maximum $sumSim(r_i.B_{head})$.

If an acronym exists for any entity reference in c_k, we compute the acronym attribute $ar_k A_{acro}$ for the authority record ar_k and concatenate it with the heading, providing more information about the entity. Analogous to Equation (2), it is computed as follows:

$$ar_k.A_{acro} = r_i.B_{acro}$$

such that $r_i.B_{acro}$ has the maximum $aCount(r_i.B_{acro})$.

Configuration of the WER Framework

We implemented a generic and configurable framework for entity resolution and creation of authority files. The framework is composed of classes corresponding to the steps of WER. These classes contain basic operations for generic entity resolution and creation of authority files, and they can be extended for domain-specific applications.

The WER framework is configurable and the user can, for example, define the attributes of the entity records, select the attributes to compose the queries, the attribute corresponding to the primary entity, the type of document to be collected (text snippet or full text), the similarity function to be used to compare strings in the extraction of information and in the clustering procedure (e.g., edit distance, the Jaccard coefficient, and cosine), the clustering procedure to be used, and the weights to be used on the clustering similarity function.

The WER framework also implements some clustering metrics to evaluate experimental results, such as the *K* metric and pairwise F1 (Pereira et al., 2009). It is useful for simulations, allowing to model different strategies for entity resolution and to measure their results.

Computational Complexity

The computational time complexity of our method without any improvement is $O(n^2)$, where n is the number of records in the input set. This complexity is dominated by the similarity function that compares each pair of records. To improve the scalability of our method, we employ blocking techniques, such as in Jaro (1989), McCallum, Nigam,

DOI: 10.1002/asi

and Ungar (2000), On, Lee, Kang, and Mitra (2005). The goal of blocking is to group similar inputs into the same block by some criteria, so that we can apply the pairwise similarity function and the clustering procedure per block. The computational time complexity after blocking becomes $O(n + b * m^2)$, where b is the number of blocks and m is the average number of records per block. In general, $b * m^2 \ll n^2$.

To demonstrate the scalability of the improved version of our method, we performed an experiment of blocking using the DBLP dataset[1] (collected in March, 2010), composed of 4,290,207 author name strings. All names with the same first name initial and the same last name were grouped into the same block. This heuristic generated 445,333 blocks, with an average number of 9.6 author names per block, and took just 20 seconds to run in a core 2 duo processor with 4 GB of RAM. Considering the time to compare each pair of records in our similarity function, as described in the next section, our algorithm requires close to 5 minutes to pairwise compare all records in the DBLP dataset.

In terms of the number of Web accesses, our method is $O(n)$, since we submit n queries to the search engine. This is the most time-consuming phase in our approach. Using the Google Search API (Google API, 2009), we submitted and collected the answer set of about 83,000 queries per day, using a single machine. To improve even further the scalability of the algorithm, we can distribute the collection into several machines, each one with a distinct network proxy.

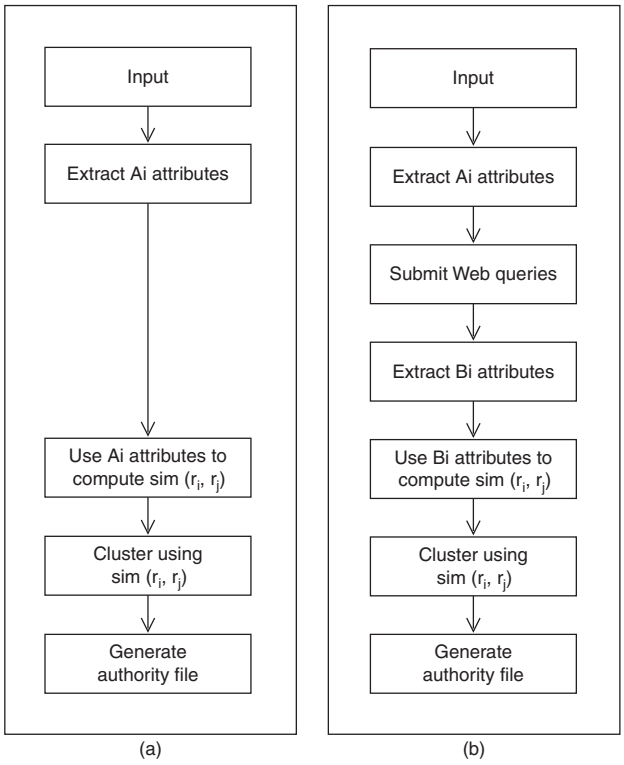

FIG. 2. Algorithms used in our experiments: (a) the baseline, which computes $sim(r_i, r_j)$ using just the domain-specific (A_i) attributes and (b) WER, which computes $sim(r_i, r_j)$ using just the Web-based (B_i) attributes.

Experimental Setup

In this section, we present our experimental evaluation of the WER method. Figure 2 illustrates the design of the experiments. The objective is to demonstrate the effectiveness of using the Web-inferred attributes B_i to guide the clustering procedure. For this, we compare the results produced using just the B_i attributes with those produced using only the domain-specific attributes A_i.

To compare attributes composed of strings, we apply the Jaccard similarity coefficient metric (Jaccard, 1901). We experimented with other similarity functions such as edit distance, cosine, and Jaccard with edit distance (French et al., 2000), but the differences in the results were minimal.

For clustering data, the baseline and WER methods were executed using a KNN clustering technique, analogous to the first phase of Karypis et al.'s method (Karypis, Han, & Kumar, 1999). Given that the similarity between record r_i and record r_j is $sim(r_i, r_j)$, as specified in Equation (3), we define a KNN graph representation, as follows. With each entity record r_i is associated a node in the graph. Given r_i, consider all records r_j for which the similarity $sim(r_i, r_j)$ is higher than a given threshold. Among these, take the K records that have the highest similarity values—these are the KNNs of record r_i. Between each node in the set of KNN neighbors and record r_i, create an edge connecting them. The resultant graph is the KNN graph. Each connected component in this graph is a cluster that represents an entity of the world.

To empirically tune the parameters and the linear weight combination of Equation (3), we used a cross-validation technique with five random sub-samples, i.e., we run WER 5 times, each time using 50% of the data, selected at random, for training and the remainder 50% for test. The best parameters found in the training data (with exhaustive search) are applied to the test set. We also applied the same strategy for the baseline, each time using the same sub-sample used by WER. We reported here the average of these five runs.

For obtaining information from the Web, we submitted queries using the Google Search API (Google AIP, 2009) and collected the top ten text snippets for each query.

The experiments were performed on the following three problems: printer description, publication venue title, and author name resolution. In the following sections, we present the evaluation metrics and report the experiments and results for each one of the three problems.

Evaluation Metrics

We evaluate experimental results based on clustering metrics adopted by previous works (Huang, Ertekin, & Giles, 2006; Laender et al., 2008; Song, Huang, Councill, Li, & Giles, 2007): the K metric and pairwise F1. The K metric consists of the geometric mean between average cluster purity

[1] http://dblp.uni-trier.de/xml/

DOI: 10.1002/asi

(ACP) and average entity purity (AEP), determining the equilibrium between the two metrics. ACP evaluates the purity of the generated clusters with respect to reference clusters manually generated, i.e., whether the generated clusters include only entity records belonging to the reference clusters. AEP evaluates the level of splitting of one entity into several clusters, i.e., how fragmented the generated clusters are. They are defined as

$$ACP = \frac{1}{N}\sum_{i=1}^{q}\sum_{j=1}^{R}\frac{n_{ij}^2}{n_i}$$

$$AEP = \frac{1}{N}\sum_{j=1}^{R}\sum_{i=1}^{q}\frac{n_{ij}^2}{n_j}$$

$$K = \sqrt{ACP \times AEP}$$

where N is the number of entity records in the test dataset; R the number of reference clusters (manually generated); q the number of clusters automatically generated; n_{ij} the number of elements of cluster i belonging to cluster j; and n_i the number of elements of cluster i.

Pairwise F1 (pF1) is defined as the harmonic mean of pairwise precision (pp) and pairwise recall (pr), where pairwise precision is measured by the fraction of pairwise entity records associated with the same entity in the clusters, and pairwise recall is the fraction of pairwise entity records associated with the same entity placed in the same cluster. They are defined as

$$pp = \frac{\sum_{i=1}^{q}\sum_{j=1}^{R} C(n_{ij}, 2)}{\sum_{i=1}^{q} C(n_i, 2)}$$

$$pr = \frac{\sum_{i=1}^{q}\sum_{j=1}^{R} C(n_{ij}, 2)}{\sum_{j=1}^{R} C(n_j, 2)}$$

$$pF1 = \frac{2 \times pp \times pr}{pp + pr}$$

where $C(n, r)$ is the combination of r elements from a set of n elements, $C(n, r) = n!/(r! \times (n-r)!)$, $n \geq r$.

Experimental Results of the WER Framework

In this section, we discuss results of the application of the generic WER framework to three distinct entity resolution problems: printer descriptions, publication venue titles, and author names. In all three cases, we compare the results with those produced by baselines that use only the domain-specific attributes to cluster entity records. The comparison suggests that the generic WER framework yields results of superior quality.

WER Entity Resolution Applied to Printer Descriptions

In this section, we present the experiments we performed to evaluate the WER method for the resolution of printer descriptions. This is a common problem faced by online product catalog services, which need to consolidate product descriptions extracted from different Web pages into lists associated with the same product. For instance, consider the following printer descriptions found in our dataset:

HP Officejet J3680 All-in-One Printer, Fax, Scanner, Copier
HP CB071A#A2L J3680 Officejet Multifunction Printer
Hewlett Packard Officejet 3680 All-in-One Printer (CB071A)
HP Officejet J4580 All-in-One Printer, Fax, Scanner, Copier

The three first descriptions refer to the same printer, and the last description, despite very similar to the first, corresponds to a different printer.

TABLE 1. Comparison of WER against our baseline.

Method	K (%)	pF1 (%)
WER	76.3±1.0	65.7±3.7
Baseline	53.3±1.0	29.0±1.3
Gain of WER	**43.2**	**126.6**

Each cell shows the value for the K and pairwise (pF1) metrics, and its 95% confidence interval. The last line shows the gain of WER for each metric. The gains are statistically significant for all metrics.

Dataset. For the experiments, we used a real dataset of printer descriptions obtained by querying Google's Product Search. First, we manually collected distinct printer descriptions from sites of several manufactures (HP, Epson, Canon, and others). Then, we used these printer descriptions as queries to Google's Product Search and collected the titles of each document in the answer set. Finally, we manually selected, for each query, those distinct results that really corresponded to the queried printer. The resultant dataset is composed of 2,169 distinct printer description strings, which we used for our experiments. These strings are distributed in 158 clusters, having on an average 13.7 strings per cluster.

Results. As explained before, to fine tune the parameters and the linear weight combination of Equation (3), we used a cross-validation technique with five random sub-samples. The parameters were stable in the five training sets, resulting in weights 0.5 and 0.5 for the ranking function of heading (F_{head}) and pair of URLs (F_{url}), respectively. The number of common URLs was empirically set to 3 (the q parameter), also based on experiments in the training data. For querying the Google search engine, we used the value of the description attribute.

Table 1 shows the results comparing the baseline, the Jaccard similarity coefficient, with WER, using the K and pairwise F1 (pF1) metrics. The last line in the table presents the gains of WER, which are statistically significant at a 95% confidence level for all metrics.

We manually verified the clustering results and observed that similar strings, such as "HP Officejet J3680 All-in-One Printer, Fax, Scanner, Copier" and "HP Officejet J4580 All-in-One Printer, Fax, Scanner, Copier," describing distinct printers were put together in the same cluster by the baseline

DOI: 10.1002/asi

and in distinct clusters by WER. In addition, strings with few words in common, such as "HP Officejet J3680 All-in-One Printer, Fax, Scanner, Copier" and "HP CB071A#A2L J3680 Officejet Multifunction Printer," describing the same printer were put together in the same cluster by WER and in distinct clusters by the baseline. In both cases, the URLs were important to disambiguate them.

Analyzing the WER cases of failure, we found errors in the disambiguation of printers that have variations of the same model, such as "HP LaserJet P3005 Printer," "HP LaserJet P3005d Printer," and "HP LaserJet P3005dn Printer." They have very similar strings and their queries returned some URLs in common, which made WER put them into the same cluster, even though they were distinct printers. Another case of failure occurs when the printer description contains several other abbreviated specifications besides its simple description. For instance, the printer "Lexmark x363dn Multifunction Printer" has as one of its descriptions the string "Lexmark 13b0501 × 363dn mfp mono fb adf enet usb 1200dpi 128mb" which details other printer specifications embedded in its description. Such detailed descriptions produce few or no common URL with other descriptions of the same printer, which made WER put them into distinct clusters. In both cited cases, the baseline also produces erroneous results.

Authority file. We evaluated the quality of the heading of the authority file produced by WER. We compared the heading extracted from the Web with the original printer description value. We considered good headings as those containing the printer description without abbreviations or additional specifications such as code, speed, size, and memory.

We randomly selected 108 (5%) printer descriptions from our dataset, manually evaluated their headings, and classified the results in four ways: (1) the heading is good and the original description is not, (2) the heading and the original description are both good, (3) neither the heading nor the original description is good, and (4) the heading is not good and the original description is. We obtained the following results for the four cases, respectively: 40.7, 28.7, 23.2, and 7.4%. Such results demonstrate that WER obtained a better description for 40.7% of the printers, with an error rate of 7.4%. Note that 63.9% of the input descriptions are not good to be used as a printer canonical description, and using the headings obtained by WER only 30.6% of the descriptions remain not good.

WER Resolution Applied to Publication Venue Titles

In this section, we discuss experiments we performed to evaluate the WER method for resolution of publication venue titles. This is a common problem in digital libraries, which need to identify the bibliographic records relative to the same publication venue. The problem is made more complex by the fact that publication venue titles might appear written in distinct forms in citation records. For example, some of the references to the VLDB conference found in our dataset are as follows:

VLDB
VLDB Conference
International Conference on Very Large Data Bases
International Conference on Very Large Databases
Int. Conf. on Very Large Data Bases (VLDB)

The first and the third strings have no similarity when compared using traditional string-matching techniques such as cosine or the Jaccard similarity coefficient.

Dataset. For testing, we used a real dataset of citations (bibliographic records) obtained by querying Google Scholar. It consists of Computer Science publications from four American universities (Stanford, MIT, Harvard, and UC Berkeley). We chose Google Scholar because it stores real data automatically crawled from Web pages. We used the names of faculty members as queries to Google Scholar and collected the bibtex entries in the answer set of each query. We selected only articles/papers that included information on publication venue. Google Scholar neither identifies the type of a publication venuenor converts it into a canonical form. The resultant dataset is composed of 16,689 citation records containing 8,399 distinct publication venue title strings, which we used for our experiments.

It is time consuming to manually determine the correct cluster of each of these 8,399 strings. Besides, we need to determine the publication venue canonical title and acronym for each cluster. Inspired by French et al. (2000), we measured our results based on sample test bases. After a preliminary execution of our system, we defined two sample test bases. For the first sample, we randomly selected 110 clusters. We named this sample as *sample-at-random*. For the second sample, which we named as *sample-of-the-largest*, we chose the 50 clusters with the largest support, i.e., the largest number of non-distinct citations in our input collection. For each publication venue of each sample, we manually determined its canonical title, acronym, and all strings in our input collection that represent citations to that publication venue.

The *sample-at-random* dataset has a total of 691 distinct strings, representing 8.2% of the input strings. This sample has on an average 6.3 strings per cluster, the largest cluster has 81 strings, and there are 46 single clusters. The *sample-of-the-largest* dataset has a total of 1,142 distinct strings, representing 13.6% of the input strings. This sample has on an average 22.8 strings per cluster, the largest cluster has 81 strings, and there are only three single clusters (i.e., with only one string).

Results. The parameters of Equation (3) were stable in the five training sets, for both samples, resulting in weights 0.6, 0.2, and 0.2 for the ranking function of heading (F_{head}), acronym (F_{acro}), and pair of URLs (F_{url}), respectively. The number of common URLs was empirically set to 2 (the q parameter) based on experiments run on the training data.

DOI: 10.1002/asi

TABLE 2. Comparison of WER against our baseline for the *sample-at-random* and *sample-of-the-largest* sets from our dataset.

Method	*K*	pF1
Sample-at-random (%)		
WER	84.7±1.0	77.6±1.8
Baseline	81.4±1.2	73.0±2.1
Gain of WER	**4.1**	**6.3**
Sample-of-the-largest (%)		
WER	80.8±2.3	73.7±5.3
Baseline	73.5±2.9	59.7±6.3
Gain of WER	**9.9**	**23.4**

Each cell shows the value for the *K* and pairwise (pF1) metrics, and its 95% confidence interval. The last line of each sample shows the gain of WER for each metric. The gains on *K* and pF1 are statistically significant.

For querying the Google search engine, we used the value of the publication venue title attribute.

Table 2 shows the results comparing the baseline, the Jaccard similarity coefficient, with WER, using *K* and pairwise F1 (pF1). The last line of each sample in the table presents the gains of WER, which are statistically significant at a 95% confidence level for *K* and pF1 metrics, on both the *sample-at-random* and *sample-of-the-largest* sets.

We manually verified the clustering results and observed that short and long strings, such as "*ACM SOSP*" and "*Symposium on Operating Systems Principles,*" which have no similarity between them using traditional string-matching techniques, were put together in the same cluster by WER and in distinct clusters by the baseline.

Authority file. We evaluated the quality of the heading of the authority file produced by WER. We compared the heading extracted from the Web with the original publication venue title value. We considered good headings as those containing the full publication venue title without abbreviations, acronyms, or additional information such as place of event, date, volume, and page number. Small variations within the titles were allowed, such as the inclusion of the sponsor and the use of the words "Annual" or "International." For instance, "*IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*" and "*IEEE Conference on Computer Vision and Pattern Recognition*" are both considered good headings. But "*CVPR*" and "*Comp. Vision Patt. Recog.*" are not.

We randomly selected 92 (5%) publication venue titles from our two sample datasets, manually evaluated their heading, and classified the results into four cases: (1) the heading is good and the original title is not, (2) the heading and the original title are both good, (3) neither the heading nor the original title is good, and (4) the heading is not good but the original title is. We obtained the following results for the four cases, respectively: 25.0, 57.6, 10.9, and 6.5%. Such results demonstrate that WER obtained a better title for 25.0% of the publication venues, and it had an error rate of 6.5%.

Publication venues are usually identified by their acronyms. Then, we also evaluated the quality of the acronym obtained by WER. We considered as correct the cases in which the publication venue has an acronym and was correctly obtained or in which the publication venue does not have an acronym and WER did not obtain it. In all other cases, it is considered incorrect. To compare an acronym obtained by WER with the original acronym, we considered that an original acronym exists if it can be easily identified in the original title by a human being. We manually evaluated the same 92 publication venue titles used to evaluate headings, and classified the results into four cases: (1) the acronym was correctly obtained and the original acronym does not exist, (2) the acronym was correctly obtained and the original acronym exists, (3) the acronym was not obtained and the original really does not exist, and (4) the acronym was incorrectly obtained or it was not obtained and the original acronym exists. We obtained the following results for the four cases, respectively: 51.1, 14.1, 23.9, and 10.9%. Such results demonstrate that WER produced a correct acronym for 89.1% of the cases, and it had an error rate of 10.9%.

WER Resolution Applied to Author Names

In this section, we discuss experiments we performed to evaluate the WER method for author name resolution. This is a common problem in digital libraries, which need to identify bibliographic records relating to the same author (Cota, Ferreira, Nascimento, Gonçalves, & Laender, 2010; Han, Giles, Zha, Li, & Tsioutsiouliklis, 2004; Han, Zha, & Giles, 2005; Huang et al., 2006; Kang et al., 2009; Lee, On, Kang, & Park, 2005). However, we cannot solve this problem using only author names due to the existence of multiple authors with the same name (polysemes) or different name variations for the same author (synonyms). For example, consider the following three bibliographic citations:

> c_1: H.S. Paul, A. Gupta, and A. Sharma, "Finding a Suitable Checkpoint and Recovery Protocol for Distributed Applications", Journal of Parallel and Distributed Computing, 2006.
>
> c_2: S. Karmakar and A. Gupta, "Fault-tolerant Topology Adaptation by Localized Distributed Protocol Switching", IEEE International Conference on High Performance Computing, 2007.
>
> c_3: A. Gupta, D. Nelson, and H. Wang, "Efficient Embeddings of Ternary Trees into Hypercubes", J. Parallel Distr. Comp., 2003.

The three citations have an author called "A. Gupta." However, "A. Gupta" in citations c_1 and c_2 refers to "Arobinda Gupta," professor of the Indian Institute of Technology, and "A. Gupta" in citation c_3 refers to "Ajay Gupta," professor of Western Michigan University.

Dataset. For testing, we performed experiments on a dataset extracted from DBLP[2] and used by Han et al. (2005). The dataset is composed of 8,442 citation records, with 480 distinct authors, divided into 14 ambiguous groups, as shown

[2]http://www.informatik.uni-trier.de/~ley/db/

DOI: 10.1002/asi

TABLE 3. The 14 ambiguous groups.

Name	# of Authors	# of Citations
A. Gupta	26	577
A. Kumar	14	244
C. Chen	61	800
D. Johnson	15	368
J. Lee	100	1,417
J. Martin	16	112
J. Robinson	12	171
J. Smith	31	927
K. Tanaka	10	280
M. Brown	13	153
M. Jones	13	259
M. Miller	12	412
S. Lee	86	1,458
Y. Chen	71	1,264
Total	480	8,442

Each column lists, respectively, the name label of the ambiguous group, the number of authors each name label corresponds to, and the total number of citations in the correspondent ambiguous group.

TABLE 4. Comparison of WER against the baseline.

Method	K (%)	pF1 (%)
WER	75.5±5.5	68.3±9.5
Baseline	63.0±5.2	48.9±8.5
Gain of WER	**19.8**	**39.7**

Each cell shows the mean value among the 14 groups of authors for the K and the pairwise (pF1) metrics, and its 95% confidence interval. The last line shows the gain of WER for each metric.

in Table 3. Each record contains the following attributes: ambiguous author name, coauthor names, work title, and publication venue title.

Results. The parameters of Equation (3) were stable in the five training sets, resulting in weight 1.0 for the ranking function of pair of URLs (F_{url}). The number of common URLs was empirically set to 2 (the q parameter), based on the performance in the training data. For querying the Google search engine, we used the value of the ambiguous author name and the work title attributes. The idea is that, if two distinct queries return common URLs, probably such documents contain publications of the same author.

Table 4 shows the results comparing the baseline, the Jaccard similarity coefficient, with WER, using K and pairwise F1 (pF1) metrics. For the baseline, each citation is represented by the string formed by concatenating the coauthor name, work title, and publication venue title attribute values. The last line in the table presents the gains of WER, which are statistically significant at a 95% confidence level for the K and pF1 metrics.

We manually verified the clustering results for WER and identified that many citations of distinct authors were put together in the same cluster. This happens because many documents in the answer sets of the queries contain citations of distinct authors. Such documents are, for example, text of articles or pages of articles in digital libraries that also contain their bibliographic references, or list of articles from a journal or conference edition. Probably, we could obtain better results if we identified the documents in the answer sets of the queries that contained only publication of a single author, such as his/her curricula vitae. This motivated us to implement a specific solution for author name resolution.

Authority file. We evaluated the quality of the heading of the authority file produced by WER. We compared the heading extracted from the Web with the original author name. We considered as good headings those containing an extended or full name of the author. For example, "Arobinda Gupta" is an extended name for the abbreviated name "A. Gupta."

We randomly selected 84 (1%) author name references from our dataset, manually evaluated their headings based on the DBLP author page or on online curricula vitae of the authors, and classified the results into four cases: (1) the heading is a correct extended name and the original name is abbreviated, (2) the heading and the original name are both expanded and correct, (3) the heading and the original name are both abbreviated and correct, and (4) the heading is incorrect. We obtained the following results for the four cases, respectively: 17.9, 9.5, 67.9, and 4.7%. Such results demonstrate that WER obtained a correct expanded name for 27.4% of the authors, and it had an error rate of 4.7%.

Specialized and Generic Solutions to the Resolution of Author Names

A specialized solution for disambiguating entities may be implemented using specific knowledge about the application domain. In previous work (Pereira et al., 2009), we implemented a specialized solution for author name resolution, whose results we now compare with those produced using the WER framework.

As illustrated in Figure 3, the specialized solution preserves similarities to the generic WER solution in the following points: (a) it submits queries to a Web search engine using the attribute values of the citation records, (b) it extracts attributes from the documents in the answer set, and (c) it applies a clustering procedure to group citations of the same author. However, the specialized solution is distinct in that it uses knowledge about the application to extract the following attributes from the documents: URL, IHF, and single author. URL is used to identify the citations that each document contains. IHF (*inverse host frequency*) (Tan, Kan, & Lee, 2006) is used to quantify the relative rarity of the Internet host of the document. And single author indicates whether a document contains citations of a single person such as a curriculum vitae.

In addition, our specialized solution uses an HAC procedure that groups citations found in single author documents with high IHF. Such documents are probably curricula vitae or home pages containing citations of a single person. For those single author documents with low IHF, WER groups only those citations that are also found together in other documents. Such documents are probably pages of authors in digital libraries, and WER looks for more evidence to group

DOI: 10.1002/asi

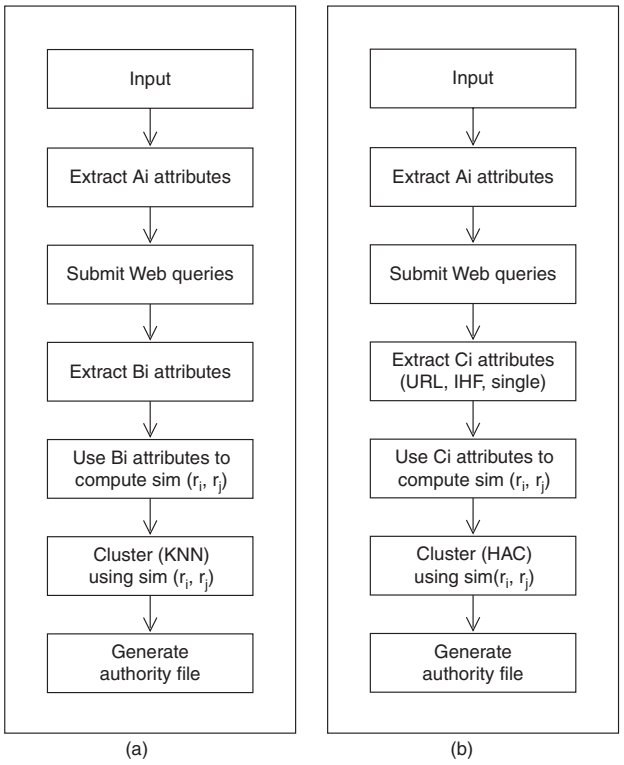

FIG. 3. Comparison between (a) WER generic solution and (b) WER-specific solution for author name resolution. The differences are in the Web-inferred attributes and in the clustering procedure.

TABLE 5. Comparison of our specialized solution (WER specific) against our generic one (WER generic).

Method	K (%)	pF1 (%)
WER (specialized)	82.4±3.7	80.1±7.1
WER (generic)	75.5±5.5	68.3±9.5

Each cell shows the mean value among the 14 groups of authors for the K and pairwise (pF1) metrics, and its 95% confidence interval.

their citations, since digital libraries may contain errors. Greater details can be found in Pereira et al. (2009).

Table 5 shows the comparison of our specific solution (WER specific) against our generic one (WER generic). As we can see, the solutions are statistically tied under all metrics. This shows that our generic solution achieves good results when compared with a specialized solution (one which cannot be directly applied to other domains).

Related Work

The entity resolution problem has been studied in various disciplines under different names such as record linkage (Fellegi & Sunter, 1969), identity uncertainty (Pasula, Marthi, Milch, Russell, & Shpitser, 2002), citation matching (Lee, Kang, Mitra, Giles, & On, 2007), merge/purge (Hernández & Stolfo, 1995), deduplication (de Carvalho, Gonçalves, Laender, & da Silva, 2006), and others. Elmagarmid, Ipeirotis, and Verykios (2007) present a survey on entity resolution.

Works on entity resolution aim at identifying entity references judged to represent the same real-world entity. In some cases, the goal is to identify replicated references, in which the attribute values differ syntactically (Bilenko et al., 2003; Sarawagi & Bhamidipatty, 2002), like identifying customer entities in customer databases coming from different subsidiaries of a company. In other cases, the goal is to identify references to the same entity, but the references do not represent replicas (Bhattacharya & Getoot, 2007; Cohen, 1998), like allocating bibliographic citations to the corresponding author. The citations allocated to each author are not necessarily replicas. In addition, after identifying replicas, some works deal with the problem of merging entity references generating a canonical form for each entity (Benjelloun et al., 2009; Wick et al., 2009). And some works also obtain an authority name for each entity (French et al., 2000; Pereira et al., 2008). In this work, we do not propose solutions to merge replicated references, but generate authority files by selecting an authority name for each disambiguated entity.

Traditional approaches to entity resolution compare entity references measuring the similarity of their attribute values based on a distance metric such as edit distance or cosine. Several works have studied algorithms for approximate string matching (Cohen, Ravikumar, Fienberg, 2003; French et al., 2000; Lawrence, Giles, & Bollacker, 1999), which can be used for entity resolution. Some approaches use supervised algorithms that learn string similarity measures from labeled data (Bilenko et al., 2003; de Carvalho et al., 2006; Tejada et al., 2001). Supervised machine-learning techniques achieve high accuracy, but require laborious human labeling and expensive training time. Our method uses an unsupervised approach, meaning that it does not require training data, but the best weighted combination needs to be determined empirically.

A recent line of work has taken the relationship among entity references into account. Dong, Halevy, and Madhavan (2005) proposed a generic framework based on a dependency graph that exploits the association among entity references belonging to multiple related classes. First, their method uses the relationship among references to provide evidence for reconciliation decisions. Then, it propagates information between reconciliation decisions for different pairs of references, and after, it addresses the lack of information in each reference by using the enriched information already obtained in the previous steps. Bhattacharya and Getoor (2007) resolve entities collectively based on the relationship among entity references, as in the bibliographic domain where author names in articles are connected by coauthor links. They use a greedy agglomerative clustering algorithm where, at any stage, the current set of clusters reflects the current belief about the mapping of the references to entities. The similarity between two clusters is dynamic, reflecting the relationship among the entity references in the clusters.

Considering efficiency and scalability issues, Benjelloun et al. (2009) proposed a generic entity resolution framework that views the functions used to compare and merge records as black-boxes, and they developed strategies that minimize the

DOI: 10.1002/asi

number of invocations to such black-boxes. Several works follow the theory suggested by Fellegi and Sunter (1969), in which efficiency is achieved by some type of blocking scheme that reduces the number of entity reference pairs to be compared. The idea is to split entity references into buckets (Jaro, 1989) or canopies (McCallum et al., 2000), and to compute similarities only among references in the same block. Bilenko, Kamath, and Mooney (2006) proposed a mechanism to automatically learn the optimal blocking function. Given similarity predicates on different entity reference attributes, their algorithm finds a nearly optimal combination of those predicates as the blocking function. The problem with blocking strategies is that an error in the blocking leads to related entity references never being put together. Other works improve the efficiency by increasing the parallelism of the entity resolution (Benjelloun et al., 2007; sik Kim & Lee, 2007). Our work does not focus on any specific solution for efficiency. Instead, we have focused on improving effectiveness.

Our work uses the Web as a source of additional information for entity resolution, as well as Elmacioglu, Kan, Lee, and Zhang (2007). Their proposal uses a Web search engine to measure how frequently two entity references appear together with the same information on the Web. If this frequency is high, an entity reference is considered a duplicate of the other. Kan and Tan (2008) examine the problem and the solutions for entity resolution in digital library metadata. They point that the Web is a fruitful source of evidence for entity resolution if carefully cleaned and utilized.

Several works aim at taking as inputs unstructured Web documents and disambiguating entities in the text of these documents linking them to named entities in a catalog, such as Wikipedia (Bunescu & Pasca, 2006; Kulkarni, Singh, Ramakrishnan, & Chakrabarti, 2009; Mihalcea & Csomai, 2007; Milne & Witten, 2008). These works differ from ours mainly in two points: (1) the inputs are unstructured texts, whereas our inputs are structured records containing domain-specific attributes and (2) the entities to be discovered are limited to those occuring in catalog or encyclopedia, whereas in our work we do not know a priori which are the entities to be discovered. Moreover, such entities are not usually all present in a previously existing list, e.g., in a catalog or encyclopedia such as Wikipedia.

Digital libraries usually maintain authority files and work hard to keep them consistent over time. One of the main initiatives in this area is the Virtual International Authority File (VIAF) project (VIAF, 2010), which combines the authority files from several of the world's national bibliographic agencies into a single authority service, available on the Web. VIAF focuses on author name disambiguation and uses domain-specific knowledge (such as author birth/death dates and ISBN numbers) to attain high precision. This is distinct from the more generic problem we try to solve in this work, one in which the solution is not fine-tuned a priori to a specific domain or target attribute.

Bennett, Hengel-Dittrich, O'Neill, and Tillett (2006) describe the initial creation of VIAF and the feasibility of algorithmically linking personal names. The details of the automated name-matching algorithms are also described. Information extracted from bibliographic records (e.g., title, co-author names, and ISBN number) and from authority records (e.g., author name and birth date) are used to disambiguate authors. Our work can contribute to enhance the VIAF name-matching algorithms, as follows. Use the domain-specific information to submit queries to a Web search engine and obtain Web-inferred attributes. Next, use those as additional evidence to name matching in the VIAF algorithms.

Besides VIAF, other projects have investigated issues related to the link of personal names in authority files (LEAF, 2010; MacEwan, 2004). Snyman and van Rensburg (2000) discuss the effort to standardize author names using a unique number, called INSAN. Several works in the literature discuss authority control and related contexts (Hickey & O'Neil, 2006; Tillett, 2004; Xia, 2006). Accordingly, one of the focus of our work is to propose solutions for the automated creation of authority files or the improvement of existing solutions. This is also one of the the focus of French et al. (2000), who investigate the use of a number of approximate string-matching techniques and introduce the notion of approximate word matching for creating authority files. A key difference with our work is that we extend the attributes of the entity references using information found on the Web, while they use only the domain-specific attributes provided in the original references. As a result, our method also produces acronyms automatically.

Detection of name variants is not a problem restricted to bibliographic citations. Warner and Brown (2001) use data mining and various types of evidence to disambiguate names of composers, lyricists, and arrangers in the Levy Sheet Music Collection. Davis, Elson, and Klavans (2003) describe a system that locates the occurrences of named entities within a text, when given a priori a set of related names included in authority lists.

The problem of author name resolution is complex due to the existence of multiple authors with the same name or different name variations for the same author (Lee et al., 2005). Besides basic metadata from citations such as author and coauthor names, work titles and publication venue titles, some works also use additional information obtained from the Web. Kang et al. (2009) explore the net effects of coauthorship on author name disambiguation and use a Web-assisted technique of acquiring implicit coauthors of the target author to be disambiguated. They submit pairs of names as queries to Web search engines to retrieve documents that contain both author names, and these documents are scanned to extract new author names as coauthors of the original pair. Tan et al. (2006) also make use of input citations to submit queries to a Web search engine and then use the URLs in the answer sets as features to disambiguate author names, considering that common URLs returned by queries related to distinct citations may indicate that the citations are of the same author. The key difference to our work is that we exploit the content of the documents returned by the queries, while they do not. As a result, they collect more noisy results because many returned

URLs do not refer to documents containing publications or because the documents are not of a single author.

Few other works aim at disambiguating personal names using information extracted from the Web (Bekkerman & McCallum, 2005; Bollegala, Honma, Matsuo, & Ishizuka, 2008; Kalashnikov, Nuray-Turan, & Mehrotra, 2008). Such works rely mainly on external features such as social networks, link structures, and attributes extracted from full documents such as e-mail, postal address, and phone numbers to identify real individuals. This problem differs from ours because it is restricted to personal names and the input records (the Web pages) do not have explicit attributes—an algorithm needs to explicitly infer and extract them.

Conclusions and Future Work

In this work, we presented a generic framework for implementing a novel authority resolution method, which uses information extracted from the Web to improve the quality of the authority file. The experiments we conducted suggest that the method is superior to more standard methods in the literature. More importantly, the generic framework, which we refer to as WER, can be easily configured for the application to distinct domains. For instance, we discussed how to configure the WER framework for resolution of printer descriptions, publication venue titles, and author names.

We evaluated our method using real datasets collected from the Web. These datasets reflect typical entity resolution problems faced by Web catalog services and digital libraries that collect data from Web sites. In particular, all records might include errors, typos, and incorrect associations. That is, our input is not sanitized. Our results indicate that large gains in the quality of the generated authority files are obtained by using the WER framework, when compared with results produced by known entity resolution algorithms. Indeed, the experimental results with three distinct datasets led to gains in the pairwise F1 metric up to 126% when compared with the selected baselines.

There are many interesting directions we can follow for future work. We intend to investigate how to use titles and texts associated with the Web-inferred attributes to improve results. We also intend to study the impact on results of the inclusion of other Web-inferred attributes such as in-links and out-links associated with the Web pages. Additionally, we intend to experiment with supervised methods such as SVM (Vapnik, 1995) or Random Forests (Breiman, 2001) to combine weights and ranking functions using domain-specific and Web-inferred attributes in our similarity function.

Acknowledgments

This work was partially supported by the Brazilian National Institute of Science and Technology for the Web (grant MCT/CNPq 573871/2008-6) and by the InfoWeb project (grant MCT/CNPq/CT-INFO 550874/2007-0). The authors also acknowledge their individual grants and scholarships from CNPq, CAPES, and FAPEMIG.

References

Auld, L. (1982). Authority control: An eight-year review. Library Resources and Technical Services, 26, 319–330.

Bekkerman, R., & McCallum, A. (2005). Disambiguating web appearances of people in a social network. In Proceedings of the 14th World Wide Web Conference (pp. 463–470). New York: ACM Press.

Benjelloun, O., Garcia-Molina, H., Kawai, H., Larson, T., Menestrina, D., & Thavisomboon, S. (2007). D-swoosh: A family of algorithms for generic, distributed entity resolution. In Proceedings of the 27th International Conference on Distributed Computing Systems (pp. 37–46). Washington, DC: IEEE Computer Society.

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., & Widom, J. (2009). Swoosh: A generic approach to entity resolution. The VLDB Journal-The International Journal on Very Large Databases, 18(1), 255–276.

Bennett, R., Hengel-Dittrich, C., O'Neill, E.T., & Tillett, B.B. (2006, August). Viaf (virtual international authority file): Linking Die Deutsche Bibliothek and Library of Congress name authority files. Paper presented at the World Library and Information Congress: 72nd IFLA General Conference and Council, Seoul, Korea.

Bhattacharya, I., & Getoor, L. (2007). Collective entity resolution in relational data. ACM Transaction on Knowledge Discovery from Data, 1(1), 5.

Bilenko, M., Kamath, B., & Mooney, R.J. (2006). Adaptive blocking: Learning to scale up record linkage. Proceedings of the Sixth IEEE International Conference on Data Mining (pp. 87–96). Washington, DC: IEEE Computer Society.

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., & Fienberg, S. (2003). Adaptive name matching in information integration. IEEE Intelligent Systems, 18(5), 16–23.

Bollegala, D., Honma, T., Matsuo, Y., & Ishizuka, M. (2008). Mining for personal name aliases on the web. In Proceedings of the 17th World Wide Web Conference (pp. 1107–1108). New York: ACM Press.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Bunescu, R., & Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (pp. 9–16). Stroudsburg, PA: Association for Computational Linguistics.

Cohen, W.W. (1998). Integration of heterogeneous databases without common domains using queries based on textual similarity. In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (pp. 201–212). New York, NY: ACM Press.

Cohen, W.W., Ravikumar, P., & Fienberg, S.E. (2003). A comparison of string distance metrics for name-matching tasks. In Proceedings of IJCAI Workshop on Information Integration on the Web (pp. 73–78). Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Cota, R.G., Ferreira, A.A., Nascimento, C., Gonçalves, M.A., & Laender, A.H.F. (2010). An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. Journal of the American Society for Information Science and Technology, 61(9), 1853–1870.

Croft, W.B., Metzler, D., & Strohman, T. (2009). Search engines: Information retrieval in practice. Reading, MA: Addison-Wesley.

Davis, P.T., Elson, D.K., & Klavans, J.L. (2003). Methods for precise named entity matching in digital collections. In Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 125–127). New York: ACM Press.

de Carvalho, M.G., Gonçalves, M.A., Laender, A.H.F., & da Silva, A.S. (2006). Learning to deduplicate. In Proceedings of the Sixth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 41–50). New York: ACM Press.

Dong, X., Halevy, A., & Madhavan, J. (2005). Reference reconciliation in complex information spaces. In Proceedings of the 25th ACM SIGMOD International Conference on Management of Data (pp. 85–96). New York: ACM Press.

Elmacioglu, E., Kan, M.-Y., Lee, D., & Zhang, Y. (2007). Web based linkage. In Proceedings of the Ninth Annual ACM International Workshop on Web Information and Data Management (pp. 121–128). New York: ACM Press.

DOI: 10.1002/asi

Elmagarmid, A.K., Ipeirotis, P.G., & Verykios, V.S. (2007). Duplicate record detection: A survey. IEEE Transaction on Knowledge and Data Engineering, 19(1), 1–16.

Fellegi, I.P., & Sunter, A.B. (1969). A theory for record linkage. Journal of the American Statistical Association, 64(328), 1183–1210.

French, J.C., Powell, A.L., & Schulman, E. (2000). Using clustering strategies for creating authority files. Journal of the American Society for Information Science, 51(8), 774–786.

Google, A.P.I. (2009). Google ajax search api. Retrieved from http://code.google.com/apis/ajaxsearch

Han, H., Giles, C.L., Zha, H., Li, C., & Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In Proceedings of the Fourth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 296–305). New York: ACM Press.

Han, H., Zha, H., & Giles, C.L. (2005). Name disambiguation in author citations using a k-way spectral clustering method. In Proceedings of the Fifth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 334–343). New York: ACM Press.

Hernández, M.A., & Stolfo, S.J. (1995). The merge/purge problem for large databases. ACM SIGMOD Record, 24(2), 127–138.

Hickey, T.B., & O'Neill, J.T.E.T. (2006). NACO normalization: A detailed examination of the authority file comparison rules. Library Resources & Technical Services, 50(3), 166–172.

Huang, J., Ertekin, S., & Giles, C.L. (2006). Efficient name disambiguation for large-scale databases. In Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases. Lecture Notes in Artificial Intelligence, 4213, 536–544.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura [Comparative study of the floral distribution in a portion of the Alps and the Jura]. Bulletin del la Société Vaudoise des Sciences Naturelles, Vol. 37 (1901), pp. 547–579.

Jaro, M.A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association, 84(406), 414–420.

Kalashnikov, D.V., Nuray-Turan, R., & Mehrotra, S. (2008). Towards breaking the quality curse: A web-querying approach to web people search. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 27–34. New York: ACM Press.

Kan, M.-Y., & Tan, Y.F. (2008). Record matching in digital library metadata. Communications of the ACM, 51(2), 91–94.

Kang, I.-S., Na, S.-H., Lee, S., Jung, H., Kim, P., Sung, W.-K., & Lee, J.-H. (2009). On co-authorship for author disambiguation. Information Processing & Management, 45(1), 84–97.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32(8), 68–75.

Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). Collective annotation of Wikipedia entities in web text. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 457–466). New York: ACM Press.

Laender, A.H.F., Gonçalves, M.A., Cota, R.G., Ferreira, A.A., Santos, R.L.T., & Silva, A.J.C. (2008). Keeping a digital library clean: New solutions to old problems. In Proceedings of the Eighth ACM Symposium on Document Engineering (pp. 257–262). New York: ACM Press.

Larkey, L.S., Ogilvie, P., Price, M.A., & Tamilio, B. (2000). Acrophile: An automated acronym extractor and server. In Proceedings of the Fifth ACM International Conference on Digital Libraries (pp. 205–214). New York: ACM Press.

Lawrence, S., Giles, C.L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. IEEE Computer, 32(6), 67–71.

LEAF (2010). LEAF project consortium. Retrieved from http://www.crxnet.com/leaf/

Lee, D., Kang, J., Mitra, P., Giles, C.L., & On, B.-W. (2007). Are your citations clean? Communications of the ACM, 50(12), 33–38.

Lee, D., On, B.-W., Kang, J., & Park, S. (2005). Effective and scalable solutions for mixed and split citation problems in digital libraries. In Proceedings of the Second International Workshop on Information Quality in Information Systems (pp. 69–76). New York: ACM Press.

MacEwan, A. (2004). Project interparty: From library authority files to e-commerce. Cataloging & Classification Quarterly, 39(1 & 2), 429–442.

McCallum, A., Nigam, K., & Ungar, L.H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 169–178). New York: ACM Press.

Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In Proceedings of the 16th ACM Conference on Information and Knowledge Management (pp. 233–242). New York: ACM Press.

Milne, D., & Witten, I.H. (2008). Learning to link with Wikipedia. In Proceedings of the 17th ACM Conference on Information and Knowledge Management (pp. 509–518). New York: ACM Press.

On, B.-W., Lee, D., Kang, J., & Mitra, P. (2005). Comparative study of name disambiguation problem using a scalable blocking-based framework. In Proceedings of the Fifth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 344–353). New York: ACM Press.

Pasula, H., Marthi, B., Milch, B., Russell, S., & Shpitser, I. (2002). Identity uncertainty and citation matching. Paper presented at Proceedings of the Advances in Neural Information Processing Systems (pp. 1401–1408). Cambridge, MA: MIT Press.

Pereira, D.A., Ribeiro-Neto, B., Ziviani, N., & Laender, A.H.F. (2008). Using web information for creating publication venue authority files. In Proceedings of the Eighth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 295–304). New York: ACM Press.

Pereira, D.A., Ribeiro-Neto, B., Ziviani, N., Laender, A.H.F., Gonçalves, M.A., & Ferreira, A.A. (2009). Using web information for author name disambiguation. In Proceedings of the Ninth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 49–58). New York: ACM Press.

Sarawagi, S., & Bhamidipaty, A. (2002). Interactive deduplication using active learning. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 269–278). New York: ACM Press.

sik Kim, H., & Lee, D. (2007). Parallel linkage. In Proceedings of the 16th ACM Conference on Information and Knowledge Management (pp. 283–292). New York: ACM Press.

Snyman, M.M.M., & van Rensburg, M.J. (2000). Revolutionizing name authority control. In Proceedings of the Fifth ACM International Conference on Digital Libraries (pp. 185–194), San Antonio, TX.

Song, Y., Huang, J., Councill, I.G., Li, J., & Giles, C.L. (2007). Efficient topic-based unsupervised name disambiguation. In Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 342–351). New York: ACM Press.

Tan, Y.F., Kan, M.-Y., & Lee, D. (2006). Search engine driven author disambiguation. In Proceedings of the Sixth ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 314–315). New York: ACM Press.

Tejada, S., Knoblock, C.A., & Minton, S. (2001). Learning object identification rules for information integration. Information Systems, 26(8), 607–633.

Tillett, B.B. (2004). Authority control: State of the art and new perspectives. Cataloging and Classification Quarterly, 38(3–4), 23–41.

Vapnik, V.N. (1995). The nature of statistical learning theory. New York, NY: Springer.

VIAF (2010). VIAF: The virtual international authority file. Retrieved from http://viaf.org/

Warner, J.W., & Brown, E.W. (2001). Automated name authority control. In Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 21–22). New York: ACM Press.

Wick, M., Culotta, A., Rohanimanesh, K., & McCallum, A. (2009). An entity based model for coreference resolution. In Proceedings of the Ninth Society for Industrial and Applied Mathematics (SIAM) International Conference on Data Mining (pp. 365–376). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Xia, J. (2006). Personal name identification in the practice of digital repositories. Program: Electronic Library and Information Systems, 40(3), 256–267.