Entity Resolution in Large Patent Databases: An Optimization Approach

Emiel Caron and Ekaterini Ioannou
Department of Management, Tilburg University, Tilburg, The Netherlands

Keywords: Entity Resolution, Data Disambiguation, Data Cleaning, Data Integration, Bibliographic Databases.

Abstract: Entity resolution in databases focuses on detecting and merging entities that refer to the same real-world object. Collective resolution is among the most prominent mechanisms suggested to address this challenge since the resolution decisions are not made independently, but are based on the available relationships within the data. In this paper, we introduce a novel resolution approach that combines the essence of collective resolution with rules and transformations among entity attributes and values. We illustrate how the approach's parameters are optimized based on a global optimization algorithm, i.e., simulated annealing, and explain how this optimization is performed using a small training set. The quality of the approach is verified through an extensive experimental evaluation with 40M real-world scientific entities from the Patstat database.

1 INTRODUCTION

Entity Resolution (ER) is a fundamental task for data integration, cleaning, and search. It aims at detecting *entities*, also referred to as instances, profiles, descriptions, or references, that provide information related to the same real-world objects. Such entities are then merged together. For example, we can merge entities together that provide information about a particular real-world event, a location, an organization, or a person.

This paper focuses on the resolution of large collections of structured data. The processing combines the essence of collective resolution with rules among entities attributes and values. Thus, instead of just following relations among entities as done by traditional collective resolution methods, use the rules related to the particular entities. To the best of our knowledge, this is the first technique that investigates such a combination. The method also illustrates that it is possible to optimize the overall resolution performance by incorporating a global optimization algorithm by using *simulated annealing*.

The introduced method starts by pre-cleaning the entities and extracting values. Next, it constructs rules based on these values and make use of the tf-idf algorithm to compute string similarities. It then creates clusters of entities by means of a rule-based scoring system. Finally, it perform precision-recall analysis using a golden set of clusters and optimizes the parameters of the algorithm.

The contributions of this paper are outlined as follows:

- We advocate a novel generic approach that investigates collectivity for resolution using rules among attributes and values. The approach is capable to operate over collections of very large volumes.
- The approach's parameters are optimized based on a global optimization algorithm and using a tiny percentage of the collection instances for the training part.
- We evaluate quality using real-world scientific references, i.e., Patstat database with 40M instances and a high number of entities describing the same entity.

The remainder of this paper is structured as follows. In Section 2, we briefly discuss related work. Section 3 defines the problem and introduces our method for entity resolution as well as the parameter optimization approach. Section 4 presents and analyzes the results of the experimental evaluation. Section 5 provides conclusions and discusses future directions.

2 RELATED WORK

During the last decades a plethora of ER techniques have been proposed. Each of these techniques introduce mechanisms to handle particular data challenges and/or environment characteristics. As discussed in

Caron, E. and Ioannou, E.
Entity Resolution in Large Patent Databases: An Optimization Approach.
DOI: 10.5220/0010527501480156
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 148-156
ISBN: 978-989-758-509-8

recent surveys Papadakis et al. (2021); Dong and Srivastava (2015), the primary focus of the introduced techniques was on collections with an increasing data volume using pay-as-you-go mechanisms, e.g., Papenbrock et al. (2015); Wang et al. (2016), or on collections with unstructured data with high levels of heterogeneity, e.g., Papadakis et al. (2011, 2013).

Although the need for handling huge collections of data with high levels of heterogeneity is clear, a large portion of current applications are still using structured data with relatively small amounts of updates. Scientific collections are the most common example used in a plethora of research publications, i.e., data describing publications and authors. The primary ER challenges in such collections are related to duplicates and noise present in the values of the instances. The typical setting is having a single collection, usually of a very large size, in which a number of entities corresponds to the same real-world object. For example, the Worldwide Patent Statistical Database (European Patent Office, 2019) contains 40M entities and has a large number of entities describing the same real-world object (i.e., around 80).

One mechanism found successful for ER focused on *rules among the entities attributes and/or values*, also known as mappings, transformations, and correspondences Yan et al. (2001); Tejada et al. (2002). For example, *Active Atlas* Tejada et al. (2002) starts with a collection of generic transformations (e.g., abbreviation for transforming "3rd" to "third", acronym for transforming "United Kingdom" to "UK") and learns the weight of each transformation given a particular application domain.

Another mechanism that was proven to be very successful is *collective resolution* Rastogi et al. (2011); Ioannou et al. (2008); Dong et al. (2005); Kalashnikov et al. (2005). The idea here is to leverage the global information (i.e., collective) in order to detect pairs with low matching similarity (likelihood) and infer indirect matching relations through relationships detected during the processing. For collections with scientific data this typically means propagating information between detected pairs of authors and publications to accumulate additional evidences, i.e., positive and negative.

In supervised ER approaches, a data set with labelled records is used to train the learning scheme. However, large data sets with manually labelled records would be expensive to collect and are often not available. For this reason, research has focused on developing unsupervised approaches for ER. Unsupervised approaches use similarity metrics and clustering algorithms to find clusters of name variants. While unsupervised approaches do not require training data sets, they often perform less well than supervised approaches (Levin et al., 2012). In this paper, we therefore adopt a hybrid approach, where we use an unsupervised approach for clustering and improve the model result's on limited training set, reflecting a form of weak supervision.

3 OPTIMIZATION METHOD FOR ENTITY RESOLUTION

We now introduce our approach. We first provide the formal description of the problem (Section 3.1), then discuss the optimization of the parameters (Section 3.2), and finally describe the resolution algorithm (Section 3.3).

3.1 Problem Definition & Notation

Our optimization problem for ER is expressed in the following notation. Consider N entities, or representations of entities, $r_1, ..., r_N$, where each entity r_i has features. For $i = 1, ..., N$, the entity r_i has feature vector f^i consisting of M features $f^i = (f^i(1), ..., f^i(M))$. Every feature $f^i(j)$ has a domain F_j, i.e. $f^i(1) \in F_1, ..., f^i(M) \in F_M$ which is independent of i.

We label two entities *similar* if their corresponding feature vectors are similar. Stated more precisely, we define

$$\sigma_l : F_l \times F_l \rightarrow \mathbb{R}^+ \cup \{0\}, \tag{1}$$

such that $\sigma_l(f^i(l), f^j(l))$ measures the similarity of any pair $(f^i(l), f^j(l)) \in F_l \times F_l$, where $i \neq j$, and we define a distance function

$$d(f^i(l), f^j(l)) = \frac{1}{\sigma_l(f^i(l), f^j(l))}.$$

The degree to which r_i and r_j are different is presented in the vector S:

$$S(r_i, r_j) = (d(f^i_1, f^j_1), d(f^i_2, f^j_2), ..., d(f^i_M, f^j_M)).$$

To obtain a similarity score as a single number, a NxN matrix $\mathbf{S}$ is defined with elements s_{ij} and weight vector $\mathbf{w} = (w_1, ..., w_M)$:

$$s_{ij} = \sum_{k=1}^{M} w_k \cdot d(f^i(k), f^j(k)),$$
$$\text{where } w_k \geq 0, i \neq j, \text{and } \sum_{k=1}^{M} w_k = 1. \tag{2}$$

Entities that are similar, are grouped into sets, or clusters, in the following way. Suppose we have Q sets

$\Sigma_1, \Sigma_2, \ldots, \Sigma_Q$, and have defined an initial threshold δ, then

$$\cup_{p=1}^{Q} \Sigma_p = \{r_1, \ldots, r_N\} \quad (3)$$

and

$$r_i \in \Sigma_p, r_j \in \Sigma_p \text{ implies } d_{ij} \leq \delta, i \neq j \quad (4)$$

unless $|\Sigma_p| = 1$. The sets $\Sigma_1, \ldots, \Sigma_Q$ are not necessarily disjoint but are chosen to be maximal, i.e., we only consider solution of (3) and (4) that have the following property:

$$\forall_{r \in \Sigma} d(r', r) \geq \delta \implies r' \in \Sigma. \quad (5)$$

If we define a undirected graph with vertices r_i and r_j, an edge between r_i and r_j, and if $d(r_i, r_j) \geq \delta$ then the sets $\Sigma_1, \ldots, \Sigma_Q$, satisfying (3), (4), and (5) are determined with algorithms that compute the graph's connected components (Bondy and Murty, 1976) or maximal cliques (Bron and Kerbosch, 1973). Note that the solution depends on w and δ and is not unique but depends on the order of merging, i.e. selection of r_i. The weights $\mathbf{w}$ and δ are chosen such that there is an optimal match with a test 'golden sample' H of sets $\Delta_1, \Delta_2, \ldots, \Delta_p$. The golden sample H is a test sample with verified sets of entities.

The method's performance is evaluated using precision and recall analysis. The F1-score is the harmonic mean of precision and recall (Fawcett, 2006). Since the objective is to obtain sets with both high precision and high recall, we maximize the F1-score. Therefore we select the parameters $\mathbf{w} = (w_1, ..., w_M)$ and δ in such a way that

$$\mathbf{w}^*, \delta^* = \arg\max_{\mathbf{w}, \delta} L(\mathbf{w}, \delta) \quad (6)$$

where our objective function $L(\mathbf{w}, \delta)$ is the total average F1-score of the optimal match of H with L. Optimizing the method's parameters is done using a simulated annealing algorithm (Xiang et al., 1997), which is discussed next.

3.2 Optimization

In Eq. (6) the objective function is nonlinear and yields many local optima. The number of local optima typically increases exponentially as the number of variables increases (Erber and Hockney (1995)), here represented by $\mathbf{w}, \delta$. For this reason, the optimization problem cannot be solved straightforwardly with linear programming, and a global optimization method is needed such as, the simulated annealing algorithm, in order to find a global optimum, instead of getting trapped in one of the many local optima that might appear.

Maximizing the F1-score with respect to $\mathbf{w}, \delta$ in Eq. (6) is a combinatorial optimization problem. Research in combinatorial optimization focuses on developing efficient techniques to minimize or maximize a function of many independent variables. Since solving such optimization problems exactly would require a large amount of computational power, heuristic methods are typically used to approximate optimal solution. Heuristic methods are typically based on a iterative improvement strategy. That is, the system starts in a known configuration of the variables. Then some rearrangement operation is applied until a configuration is found that yields a better value of the objective function. This configuration then becomes the new configuration of the system and this process is repeated until no further improvements are found. Since this method only accepts new configurations that improve the objective function, the system is likely to be trapped in a local optima. This is where simulated annealing plays its part in the method.

The simulated annealing algorithm is inspired by techniques of statistical mechanics which describe the behavior of physical systems with many degrees of freedom. The simulated annealing process starts by optimizing the system at a high temperature such that rearrangements of parameters causing large changes in the objective function are made. The "temperature", or in general the control parameter, is then lowered in slow stages until the system freezes and no more changes occur. This cooling process ensures that smaller changes in the objective functions are made at lower temperatures. The probability of accepting a configuration that leads to a worse solution is lowered as the temperature decreases (Kirkpatrick et al., 1983). To optimize Eq. (6) the `dual_annealing()` function of the SciPy library in Python (SciPy.org, 2021) is used. This implementation is derived from the research of Xiang et al. (1997).

3.3 Resolution Algorithm

In this section the optimization problem is captured in a practical general method for ER. This method is inspired by (Caron and Eck, 2014; Caron and Daniels, 2016). An overview of the method and its main steps is presented in Figure 1. The method's inputs are N entities with features f^i, i.e. the raw data related to an ER-problem, and a golden sample H. The method's output is a set of parameters $\mathbf{w}^*, \delta^*$ that produce optimized sets $\Sigma_1, ..., \Sigma_Q$, that represent clusters of name variants.

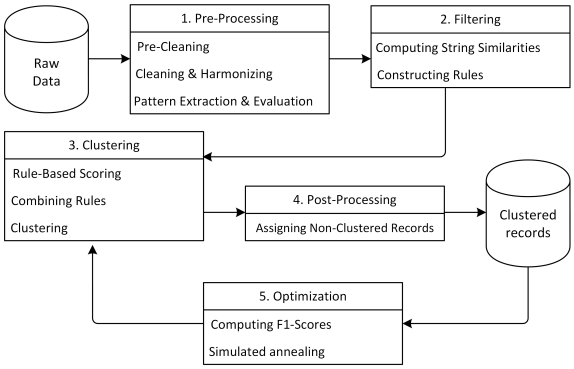

Figure 1: Graphical overview of the ER method.

The method consists of five generic steps:

1. *Pre-processing*;
2. *Filtering*;
3. *Rule-based scoring and Clustering*;
4. *Post-processing*;
5. *Optimization*.

In the method first steps $1-4$ are executed in succession on the sample data, after that equation (6) is optimized in step 5. Therefore, steps $3-5$ are executed in iteration until the total F1-score is maximized. After the final iteration the parameters $\mathbf{w}^*, \delta^*$ are obtained. With these parameters the steps $1-4$ are applied again on the whole data set to disambiguate all records. This final part is typically executed offline. In the remainder of this section we describe the steps in a concise way to give overview of general tasks within each step. Typically, these tasks need to be configured for the practical ER-problem at hand, e.g. company name disambiguation, (author) name resolution, or the cleaning of scientific references, and so on.

(1) *Pre-processing*. The method starts by preprocessing the data. During this step the entities N in raw data are pre-cleaned and harmonized to reduce basic variability in the data on the record level. In addition, the set of features M is completed, with additional descriptive labels by using techniques like regular expressions combined with match and replace actions. Finally, the feature set is evaluated for correctness.

(2) *Filtering*. The feature set M is the input for this step. Here rules are based on (combinations of) the features of the entities, and often combined with string similarity measures, e.g. the tf-idf algorithm (Salton and Buckley, 1988),to start finding similar pairs of entities. With the rule set, i.e. the input for the distance function, similarities are computed between entities using Eq. (1). By constructing rules based on the features, 'proof' is collected for the similarity between records and based on that candidate entity pairs are created. The output of this step is a pool of candidate pairs for further evaluation, where pairs that do not match are filtered out.

(3) *Rule-based scoring and clustering*. In this step, for the set of candidate pairs, initial weights $\mathbf{w}_0$ are assigned to the rules based on 'the strength of each rule', to determine Eq. (2). Typically, the strength of rules are first based on domain knowledge, after that the weights are optimized in iteration in step 5. Furthermore, the total score, i.e. the combined weights, for every pair is stored in a dataframe and compared with an initial threshold δ_0 to obtain the sets in Eq. (3). The clusters of name variants are now determined with the connected-components or the maximal cliques algorithm.

(4) *Post-processing*. In this step, entities for which no duplicates are identified in the previous step are assigned to new single-record clusters.

(5) *Optimization*. In this step, the sets are evaluated on the golden sample H using precision and recall analysis. The parameters $\mathbf{w}, \delta$ are adjusted to achieve higher values in precision and recall. Using simulated annealing (Xiang et al., 1997) we obtain the optimal parameters $\mathbf{w}^*, \delta^*$. Here the average F1-score over all sets is used, i.e. the harmonic mean of the precision and recall, as the objective function (Fawcett, 2006) defined in Eq. (6).

4 CASE STUDY EVALUATION

We now present the results of our experimental evaluation. The focus was on investigating the quality of return results as well as the effects of the introduced parameter optimization. The following paragraphs present the evaluation settings (Section 4.1), also illustrating the challenges of entity resolution of the particular data collection. After that, the results of the evaluation are analyzed (Section 4.2), followed by the details of the implementation (Section 4.3).

4.1 Setting

For the experimental evaluation, we use the Worldwide Patent Statistical Database (Patstat) (European Patent Office, 2019). This is a product of the European Patent Office designed to assist in statistical research into patent information. One of the available tables, namely *TLS214*, holds information on scientific references that are cited by patents. These references are collected from patent applications, in which

	npl_publn_id	npl_type	npl_biblio
1	2219025	s	CODD E F: 'A relational model of data for large shared data banks', COMMUNICATIONS OF THE ACM, ASSOCIATION FOR COMPUTING MAC...
2	900347890	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6, Assoc...
3	900349892	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6, Assoc...
4	900798964	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6, Assoc...
5	900942812	a	Codd, E. F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6.
6	901369158	a	CODD, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6, Asso...
7	903049816	a	CODD, E.F.: A Relational Model of Data for Large Shared Data Banks. In: Comm. of the ACM, Vol. 13, Nr. 6, Juni 1970, S. 377-387
8	903266284	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks', Jun. 1970, Communications of the ACM, vol. 13, No. 6, pp. 377-387.
9	903575625	a	E. Codd, 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM,vol. 13, No. 6, Jun. 1970, pp. 377-387.
10	903647641	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, 13 (6) : 377-387 (1970).
11	904165665	a	Codd, 'A Relational Model of Data for Large Shared Data Banks', Communication of the ACM, vol. 13, No. 6, pp. 377-387, 1970.
12	904186834	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, 13(6):377-387 (1970).
13	904492347	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, 13 (6):377-387 (1970).
14	904905079	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, 13 (6) :377-387 (1970).
15	905697167	a	Codd, E. F., A Relational Model of Data for Large Shared Data Banks , Communications of the ACM , vol. 13, No. 6, pp. 377 387, Jun. 1970.
16	905886449	a	Codd, E.F. 'A Relational Model of Data for Large Shared Data Banks' Communications of the ACM, vol. 13, No. 6, Jun. 1970, pp. 377-387.
17	906069574	a	Codd, E. A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, vol. 13, No. 6, Jun. 1970, pp. 377 387, Jun. 1970.
18	906617456	a	Codd, E. F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, 13 (6) :377-387 (1970).
19	909147086	a	Codd, E.F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6, Assoc...
20	909380209	a	Codd, E. F., 'A Relational Model of Data for Large Shared Data Banks,' Communications of the ACM, Jun. 1970, pp. 377-387, vol. 13, No. 6.

Figure 2: A sample, i.e., 20 of 80 records, matching the exact title *"A relational model of data for large shared data banks"*.

patent applicants reference scientific papers and proceedings to acknowledge the contribution of other writers and researchers to their work.

In the 2019 Spring Edition of Patstat, the TLS214 table contained a bit more than 40 million records with scientific references. The particular table is an important point of reference for researchers that wish to study the connection between science and technology. The main issue of this data, in addition to the collection size, is that amongst the scientific references there are many name variants of publications caused by missing data, inconsistent input convention, different order of items, typos, etc. These variants make the usage and analysis of the data very difficult.

To illustrate the problem with the TLS214 table, we posed a query search for an exact title of an article. The query returned 80 records, however, there may be more records referring the same real-world object. Figure 2 illustrates the first 20 results (i.e., entities) of this query. Although every record refers to the same real-world object they are stored in different ways or simply duplicated. For example, in entity 11, Codd's initials are missing, while record 7 holds an abbreviation of the word "Communications". Besides textual differences the database treats every record as a unique entity due to the primary keys. These problems make it difficult to properly retrieve information. In order for table TLS214 to be a reliable point of reference for research, its records need to be disambiguated.

Table 1: Total number of entities (i.e., records) per golden sample GS1 and GS2.

	GS1	GS2
Journal papers	17,348	11,163
Proceedings	0	1,093
Total	17,348	12,256

4.2 Result Analysis

We now discuss the experiments and results. Our optimization method is executed over the TLS214 table with the 40M entities, i.e., entities. Our goal is to investigate quality and for this we used two golden samples that contain the expected matches among the entities. The golden sample 1 (GS1) contains 100 clusters (with 17,348 records), which refer to 100 highly cited scientific papers, based on a top 100[1]. In addition, we used golden sample 2 (GS2) with in total 12,256 records, contains references to 50 unique journal publications and 50 unique conference proceedings. Both samples are evaluated by human domain experts in order to incorporate the expected entity matches (else referred to as clusters). Typically, the patterns for proceedings are more difficult to clean because they often show more variation. As can be seen in Table 1, references to journal publications occur more often than conference proceedings in Patstat.

Figure 3 gives the distribution of the number of publication name variants per cluster in GS1. Notice that the largest clusters contain more than 1,500 name variants.

We first focused on the iterative optimization part, i.e., Step 5 of the method (Section 3.3). The method is executed on a basic laptop with the intermediate results stored and compared against the ground samples. As expected, the method gradually improves the overall F1-score until this improvement is stabilized. The latter occurs after approximately 100 hours. At that point the simulated annealing algorithm stops, and the final set of parameters $\mathbf{w}^*, \delta^*$ is obtained, and clusters of name variants are derived that maximize Equation 6.

[1]Details for the top 100 highly cited scientific papers can be found here:
https://www.nature.com/news/the-top-100-papers-1.16224

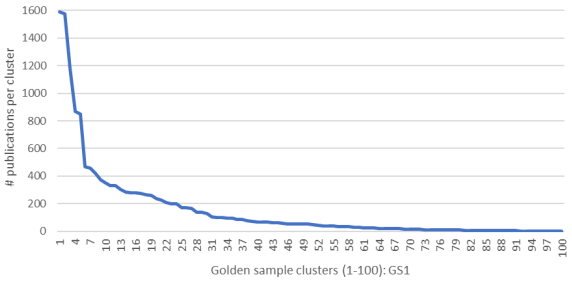

Figure 3: Distribution of the number of publication name variants per cluster in GS1, ordered descendingly.

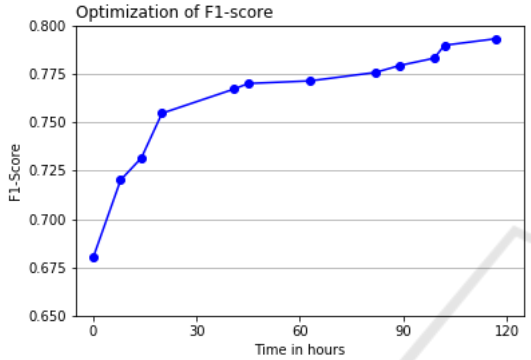

Figure 4: Simulated annealing increasing the F1-score over time.

Figure 4 shows in a plot the increase of the total average F1-score against the time in hours. The first point of the line is the F1-score of the initial clusters. This plot shows that the simulated annealing algorithm indeed is able to improve the F1-score and thus our method produces better clusters compared to the initial ones. It can be seen that the algorithm makes large improvements to the F1-score in the beginning and smaller improvements towards the end. This is in line with the theory on simulated annealing Xiang et al. (1997). The F1-score over all clusters increases from 0.68006 to 0.79304 (≈16.5%).

Table 2 shows precision-recall-F1 analysis of the final clusters for both GS1 and GS2.

Table 2: Statistics of optimized clusters for GS1 and GS2.

	GS1	GS2
Precision	0.99997	0.99361
Recall	0.92857	0.92154
F1-score	0.96295	0.95622

The table shows high average values for all performance statistics, all figures are above 0.90. Moreover, it can be observed that the average precision is slightly higher than the average recall. Obviously, the clusters obtained after optimization obtain higher precision and recall values than the initial clusters before optimization, as shown in increase of the F1-score in Figure 4.

The plots in Figure 5 give a further break-down of the improvement of precision-recall-F1 statistics on the cluster level with the optimized parameters, compared to initial parameters. The best clusters return

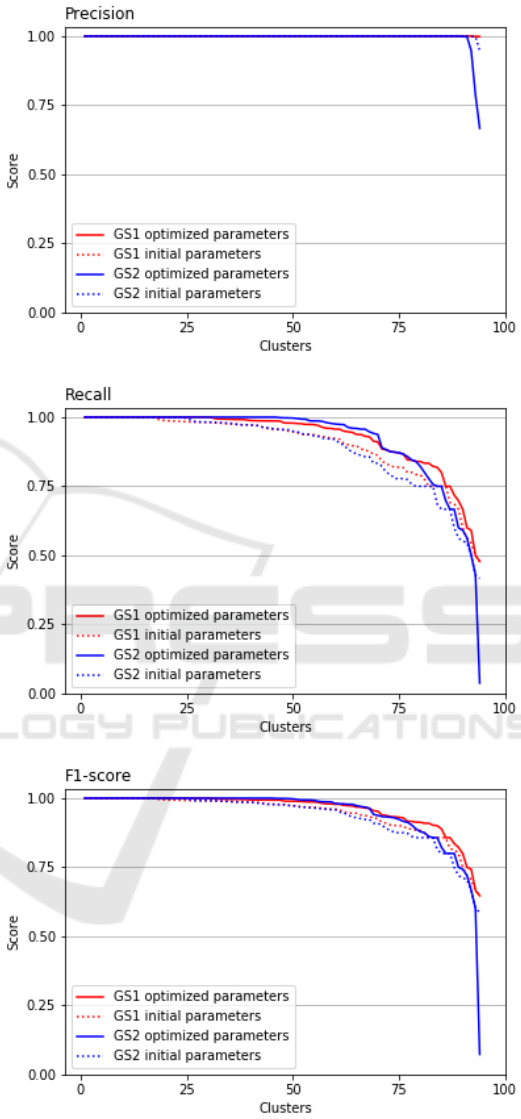

Figure 5: Distribution of the precision-recall-F1 scores of the clusters before and after optimization for GS1 and GS2.

an average precision and recall close to 100% (Table 2). The best cluster is defined as the cluster with the highest value for the F1 measure. The increase in recall indicates that the new parameters are better in creating clusters. In combination with a decrease in clusters we notice that both golden samples have larger dominant clusters after optimizing. If the three best clusters of every entity are grouped in one clus-

ter the average recall of GS1 increases from 0.92857 to 0.95084 and for GS2 from 0.92154 to 0.96397. In addition, when analyzing the clusters, we notice that our method is slightly 'conservative', it values precision over recall. As a result the method is more likely to create multiple clusters with high precision than to create one large cluster with potential errors. Typically, the method splits the name variants of one golden cluster into one large dominant cluster and multiple small clusters. The improvement in the recall in Figure 5 illustrates the difference in dominant cluster size. Based on the evaluation we conclude that with the optimized parameters the rule-based scoring system finds more evidence to cluster publications together.

The slightly lower recall results in Table 2 for GS2, might origin from the format of conference proceedings. Table 3 illustrates the problem with the format of a conference reference, it shows shows an example cluster of correctly classified scientific references. Although every reference contains the primary article information (author, title, and year) each record follows a different format. In some cases, the institute name is left out, the conference abbreviated, or contains additional information (e.g. conference date or subtitle). Due to the additional information within references to conference proceedings the disambiguation method is likely to produce some erroneous results. As a result, the rule based scoring system does not always find sufficient evidence to cluster all the name variants into one cluster.

Table 3: Different formats for conference proceedings within one cluster.

Cl.	Reference
51	D. Cohen et al., IP Addressing and Routing in a Local Wireless Network, IEEE Infocom 92: Conference on Computer Communications, vol. 2, New York (US), pp. 626 632
51	Daniel Cohen, Jonathan B. Postel, and Raphael Rom, Addressing and Routing in a Local Wireless Network, IEEE INFOCOM 1992, p. 5A.3.1-7
51	Cohen et al., 'IP addressing and routing in a local wireless network' One World Through Comminications. Florence, May 4-8, 1992, Proceedings of the conference on Computer Communications (INFOCOM), New York, IEEE, US, vol. 2, Conf. 11, May 4, 1992, pp. 626632m XP010062192, ISBN: 0/7803-0602-3
51	Danny Cohen et al.; 'IP Addressing and Routing in a Local Wireless Network'; One World Through Communications. Florence, May 4-8, 1992, Proceedings of the Conference in Computer Communications (Infocom), New York IEEE, US, vol. 2 Cof. 11, May 4, 1992
51	IP Addressing & Routing in a Local Wireless Network, Cohen et al, IEEE 92, pp. 626 632
...	...

4.3 Implementation

The resolution algorithm is implemented in the Python programming language. The code[2] is structured in the following parts:

- Pre-processing and filtering: `connection.py` and `rules.py`;
- Rule-based scoring-clustering: `rule_construction.py`, `string_matching_using_tfidf.py`, `clustering.py`, and `evaluation.py`;
- Optimization: `find_clusters.py` and `optimize.py`.

In the example case of the Patstat table with ambiguous scientific references, the entities $r_1, \ldots, r_N$ are found in the records of the table. The features of the entities are extracted bibliographic meta information, such as: publication title and year, author names, various journal information, and so on. Eq. 1 of the method refers to the rules that are developed. These rules provide evidence that two records are similar. In order to compute the string similarities for the rules we use an efficient implementation of tf-idf in Python. The scores that are assigned to the rules correspond to weight vector $\mathbf{w}$ in Eq. 2. The clustering of records is done using the connected components algorithm and results in the sets $\Sigma_1, \ldots, \Sigma_Q$ as described in the method. Specifically, the method `find_clusters()` from the script `find_clusters.py`, implements the ER method and computes the total F1-score. Its input parameters are a configuration of variables $(\mathbf{w}, \delta)$ and a table containing feature vectors. We use this method as our objective function for the `dual_annealing()` method described in the script `optimize.py`. In this way we obtain the optimal configuration of the parameters in Eq. 6.

5 CONCLUSIONS

This paper explores a novel approach for performing ER over large collections of data using a combination of collective resolution with rules between entity attributes and values. The approach uses simulated annealing algorithm to optimize the related parameters. As illustrated by the evaluation on the cleaning of scientific references in the Patstat database, the introduced optimization ER approach achieves high effectiveness without requiring a large training set, resembling approaches in weak supervised learning. To optimize the method's parameters, the overall F1-score, that is captured in the non-linear objective function, is maximized over a limited golden set of clusters. After the optimization, the obtained parameters are used to disambiguate the whole data set.

In the case study, the method is applied to the cleaning of scientific references, e.g. journal publications and proceedings, and create sets of records that

[2] https://emielcaron.nl/wp-content/uploads/2020/05/entity_resolution_optimization_code.7z

point to the same bibliographic entity. The method begins by pre-cleaning the records and extracting bibliographic labels. Subsequently, rules are developed based on the labels combined with string similarity measures, and clusters are created by a rule-based scoring system. Lastly, precision-recall analysis is performed using a golden set of clusters, to optimize the rule weights and thresholds. The results demonstrate that it is feasible to optimize the overall F1-score of disambiguation method using a global optimization algorithm, and obtain the best parameters to disambiguate the whole database of scientific reference. By changing the rules, the method can directly be applied on on similar ER-problems. Therefore, the method has a generic perspective.

In future research, several directions might be explored to obtain the optimal configuration for the method. Firstly, our method can be analyzed on other datasets for ER, to study whether the results are stable and to compare the evaluations. Secondly, this work revealed additional challenges worth investigating with respect to the incorporated rules. A possible future direction is to check the algorithm's behaviour when increasing the number of used rules, and another direction is moving towards rules that can evolve over time. Thirdly, more information is necessary about the best clustering algorithm applied to merge similar name variants, e.g. an in-depth comparison between the connected components and max-clique algorithm. Fourthly, alternative optimization techniques might be used that produce similar or even better results in terms of efficiency and/or effectiveness. A comparison between simulated annealing, Tabu search, and a genetic algorithm is therefore envisaged.

ACKNOWLEDGEMENTS

We kindly acknowledge Wen Xin Lin, Colin de Ruiter, Mark Nijland, and Prof. Dr. H.A.M. Daniels for their contributions to this work.

REFERENCES

Bondy, J. A. and Murty, U. S. R. (1976). *Graph theory with applications*, volume 290. Macmillan London.

Bron, C. and Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16:48 – 50.

Caron, E. and Daniels, H. (2016). Identification of organization name variants in large databases using rule-based scoring and clustering - with a case study on the web of science database. In *ICEIS*, pages 182–187.

Caron, E. and Eck, N.-J. V. (2014). Large scale author name disambiguation using rule-based scoring and clustering. In *Proceedings of the Science and Technology Indicators Conference*, pages 79–86. Universiteit Leiden.

Dong, X., Halevy, A., and Madhavan, J. (2005). Reference reconciliation in complex information spaces. In Özcan, F., editor, *SIGMOD*, pages 85–96. ACM.

Dong, X. L. and Srivastava, D. (2015). *Big Data Integration*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Erber, T. and Hockney, G. (1995). Comment on "method of constrained global optimization". *Physical review letters*, 74(8):1482.

European Patent Office (2019). *Data Catalog - PATSTAT Global*, 2019 autumn edition edition.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Ioannou, E., Niederée, C., and Nejdl, W. (2008). Probabilistic entity linkage for heterogeneous information spaces. In Bellahsene, Z. and Léonard, M., editors, *Advanced Information Systems Engineering, 20th International Conference, CAiSE 2008, Montpellier, France, June 16-20, 2008, Proceedings*, volume 5074 of *Lecture Notes in Computer Science*, pages 556–570. Springer.

Kalashnikov, D., Mehrotra, S., and Chen, Z. (2005). Exploiting relationships for domain-independent data cleaning. In Kargupta, H., Srivastava, J., Kamath, C., and Goodman, A., editors, *SDM*, pages 262–273. SIAM.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Levin, M., Krawczyk, S., Bethard, S., and Jurafsky, D. (2012). Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology*, 63(5):1030–1047.

Papadakis, G., Ioannou, E., Niederée, C., Palpanas, T., and Nejdl, W. (2011). Eliminating the redundancy in blocking-based entity resolution methods. In Newton, G., Wright, M., and Cassel, L., editors, *JCDL*, pages 85–94. ACM.

Papadakis, G., Ioannou, E., Palpanas, T., Niederée, C., and Nejdl, W. (2013). A blocking framework for entity resolution in highly heterogeneous information spaces. *TKDE*, 25(12):2665–2682.

Papadakis, G., Ioannou, E., Thanos, E., and Palpanas, T. (2021). *The Four Generations of Entity Resolution*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Papenbrock, T., Heise, A., and Naumann, F. (2015). Progressive duplicate detection. *TKDE*, 27(5):1316–1329.

Rastogi, V., Dalvi, N. N., and Garofalakis, M. N. (2011). Large-scale collective entity matching. *Proc. VLDB Endow.*, 4(4):208–218.

Salton, G. and Buckley, C. (1988). Term-weighting ap-

proaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523.

SciPy.org (2021). Scipy.optimize package with dual_annealing() function.

Tejada, S., Knoblock, C., and Minton, S. (2002). Learning domain-independent string transformation weights for high accuracy object identification. In *SIGKDD*, pages 350–359. ACM.

Wang, Q., Cui, M., and Liang, H. (2016). Semantic-aware blocking for entity resolution. *IEEE Trans. Knowl. Data Eng.*, 28(1):166–180.

Xiang, Y., Sun, D., Fan, W., and Gong, X. (1997). Generalized simulated annealing algorithm and its application to the thomson model. *Physics Letters A*, 233(3):216–220.

Yan, L., Miller, R., Haas, L., and Fagin, R. (2001). Data-driven understanding and refinement of schema mappings. In Mehrotra, S. and Sellis, T., editors, *SIGMOD*, pages 485–496. ACM.