ReLiC: Entity Profiling by using Random Forest and Trustworthiness of a Source - Technical Report

Shubham Varma, Neyshith Sameer, C. Ravindranath Chowdary

Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, India - 221005
shvarma@microsoft.com, nesameer@microsoft.com, rchowdary.cse@iitbhu.ac.in

Abstract. The digital revolution has brought most of the world on the world wide web. The data available on WWW has increased many folds in the past decade. Social networks, online clubs and organisations have come into existence. Information is extracted from these venues about a real world entity like a person, organisation, event, etc. However, this information may change over time, and there is a need for the sources to be up-to-date. Therefore, it is desirable to have a model to extract relevant data items from different sources and merge them to build a complete profile of an entity (entity profiling). Further, this model should be able to handle incorrect or obsolete data items. In this paper, we propose a novel method for completing a profile. We have developed a two phase method-1) The first phase (resolution phase) links records to the queries. We have proposed and observed that the use of random forest for entity resolution increases the performance of the system as this has resulted in more records getting linked to the correct entity. Also, we used trustworthiness of a source as a feature to the random forest. 2) The second phase selects the appropriate values from records to complete a profile based on our proposed selection criteria. We have used various metrics for measuring the performance of the resolution phase as well as for the overall ReLiC framework. It is established through our results that the use of biased sources has significantly improved the performance of the ReLiC framework. Experimental results show that our proposed system, ReLiC outperforms the state-of-the-art.

1 Introduction

1.1 Motivation

Entity profiling is a very challenging problem and has a large number of practical applications. Entity profiling can be used to obtain information about a real-world entity using knowledge bases or by extracting information from unstructured data sources. However, many of the publicly available data sources are not reliable and contain outdated or erroneous data. The challenge is to design a system that works well in the presence of erroneous as well as ambiguous data records. Through entity resolution, relevant data can be linked to the corresponding entity and this data can be used for various information extraction tasks on the entity. Entities can be of different types, for example, people, places, items, etc. Data having different kinds of attributes need to be used for performing entity resolution on different entities. Entity resolution is an important step for entity profiling and to the best of our knowledge, classifiers were not used in the literature for entity resolution. This motivated us to study the use of classifiers for entity resolution. Due to which, the proposed method worked effectively on different types of data. Entity profiling helps in acquiring accurate information from data extracted from various unreliable sources. But, to the best of our knowledge, assigning a bias to a reliable source for profiling was not considered earlier in the literature and this motivated us to use a biased source. The objective of this work is to develop a novel technique for effectively obtaining the complete profiles of entities using structured data records from different sources.

1.2 Introduction to Entity Profiling

Entity Profiling refers to extracting complete information about an entity using information available from different sources that may have incomplete, complete or obsolete data. The entity may be a person, place, object, organization, etc. Each of these entities has their set of attributes, and the values for these attributes are extracted from various online sources. The information provided by these sources may be incomplete, obsolete or even incorrect. Many firms collect data from different sources and create knowledge repositories on real-world entities. Knowledge bases such as Instant Checkmate[1] provide complete information about a person by using online profiles. DBPedia [1] and YAGO [34] are other knowledge bases that provide public databases on real-world entities.

Table 1. Table of queries

query	cluster	Name	Matches	Runs	Highest
q1	c1	Gavaskar		10122	
q2	c2	Amarnath	69		
q3	c3	Y Singh			169

[1] https://www.instantcheckmate.com/

Table 2. Table of records

record	Name	Matches	Runs	Highest	Source
r1	SM Gavaskar	125	10122	236	s1
r2	Lala Amarnath	24	878	118	s1
r3	Yajurvindra Singh	4	109	43	s1
r4	Gavaskar	125	10122	236	s2
r5	Mohinder Amarnath	69	4378	138	s2
r6	Yuvraj Singh	40	1900	169	s2
r7	Sunil Gavaskar	125	10122	236	s3
r8	Surinder Amarnath	10	550	124	s3
r9	Yograj Singh	1	10	6	s3
r10	Y Singh	40	1900	169	s4

Let us consider Table 1 which contains three queries. These queries are incomplete profiles and have to be completed using Table 2. Table 2 contains records along with their respective sources. For completing a profile, we must make a cluster of all records that are similar to the query. The clustering of similar records to the query is known as entity resolution.

Table 3. Table of cluster c1

record	Name	Matches	Runs	Highest	Source
r1	SM Gavaskar	125	10122	236	s1
r4	Gavaskar	125	10122	236	s2
r7	Sunil Gavaskar	125	10122	236	s3

Table 4. Table of cluster c2

record	Name	Matches	Runs	Highest	Source
r2	Lala Amarnath	24	878	118	s1
r5	Mohinder Amarnath	69	4378	138	s2
r8	Surinder Amarnath	10	550	124	s3

Table 5. Table of cluster c3

record	Name	Matches	Runs	Highest	Source
r3	Yajurvindra Singh	4	109	43	s1
r6	Yuvraj Singh	40	1900	169	s2
r9	Yograj Singh	1	10	6	s3
r10	Y Singh	40	1900	169	s4

Let us take the first query, i.e, $q1$ having its associated cluster as $c1$. $q1$ refers to the records given in Table 3. So, we associated all the records in the Table 3 to the cluster $c1$. From these records, we complete the profile for $q1$, and the completed profile is given in Table 6. Similarly, after applying entity resolution to $q3$, we get Table 5. In Table 5 we see that query can only refer to $r6$ and $r10$ based on the attribute "Highest". These two records are now used to complete the profile. Table 6 gives completed profiles. Similar assignment holds for query $q2$.

Table 6. Table of completed profiles

Query	Name	Matches	Runs	Highest
q1	Gavaskar	125	10122	236
q2	Amarnath	69	4378	138
q3	Y Singh	40	1900	169

Entity resolution is one of the key steps in entity profiling. It involves identifying all the records that refer to a particular entity from knowledge bases published by various sources. It is also referred to as the problem of record linkage, as different records relevant to a given query need to be identified. The similarity between a record and the query is computed to accomplish this task. In the final step, a query is completed by merging the information from the selected set of records.

In this paper, we present a novel technique for completing the profile of an entity by extracting information from different sources. We have used classifiers for linking records to the query. A classifier classifies each record as relevant or non-relevant corresponding to each query. The various input features and labels used for classification are discussed in Section 4.1. Section 5.2 describes the technique used for selection of an appropriate classifier model. We have also proposed a novel technique for finding trustworthiness of the sources, to deal with sources that provide erroneous or obsolete data. All the sources are assigned rating with respect to a biased source. The user selects this biased source as the most accurate and relevant source of information. For example, Wikipedia has its information contributed by the wisdom of masses and information on the wiki page needs a valid citation. From time to time, this information is updated as well as maintained. Moreover, different knowledge bases like IMDB are entirely dedicated to collect information in their domains. Therefore, it is appropriate to select them as the biased sources for their domains. Further, we have proposed a new metric for finding similarity between two records and between a record and a query. The similarity between two attribute values is dependent on their respective types. We follow different approaches for finding similarities of numerals and strings. This technique is discussed in Section 3.1. There are three major contributions in this work.

1. We have used classifiers for associating records to a query.
2. We have proposed the use of biased sources for computation of source ratings.
3. We have proposed a novel method for finding similarity between two records and between a record and a query.

The rest of the paper is organized as follows. Section 2 presents the related work. Terms and definitions are introduced in Section 3. Section 4 outlines the proposed approach. Section 5 describes the experimental results, and we conclude in Section 6.

2 Related Work

Entity profiling is a challenging and an active research problem. Li, F. et al. [24] proposed the COMET framework for entity profiling. They have assigned reliability values to every source-attribute pair. They used a two-phase method, involving confidence based matching and adaptive matching. The confidence based matching phase focuses on associating the records to queries and sources. The adaptive matching phase prunes the records linked to multiple queries for reducing the profiling error. A novel technique for profiling entities is proposed in [25], which tries to model the evolution of an entity over time. In [19], the focus is on entity profiling by considering uniqueness constraints and false values. They reduce the problem of entity profiling to a k-partite graph clustering problem by performing both global linkage and fusion simultaneously. An ontology based approach for generation of user profiles using YAGO [34] is proposed in [10].

The approaches to entity resolution can be classified into two classes - learning based and rule based approaches. A learning-based model for entity resolution using Markov Logic is proposed in [33]. FEBRL [14] (Freely extensible biomedical Record Linkage) uses SVM to learn the appropriate matching combinations. The similarity measures used for this approach is same as that in the rule-based approach proposed in [17]. MARLIN [6] (Multiply Adaptive Record Linkage with INduction) uses edit distance and cosine similarity measures along with different classifiers for measuring string similarity. A conceptual semantic framework for entity resolution is proposed in [27]. Zhao, G. et al. [41] propose a novel model for linking entities mentioned in texts to semi-structured knowledge bases, by considering relationships between entities which occur together more frequently in text. This technique has been shown to disambiguate entities effectively. Cheng, J. et al. [12] have used entity resolution for linking social networking profiles to the organisations the particular user belong to. An analysis for information extraction techniques for linking tweets to entities is has been reported in [15]. Detailed description and analysis of various entity resolution and data linkage approaches are discussed in [13,36].

An entity resolution model that focuses on an incremental approach to building clusters is presented in [18]. This model merges new records into clusters and modifies the errors based on the new updates. The two approaches proposed by authors are – connected component approach and iterative approach. These two approaches reduce linkage time without compromising on the quality. Syed, H. et al. [35] discuss refinements in matching rules for the entity resolution using a two-phase model. The first phase computes both the primary attributes of the entity and the baseline matching rules. The second phase involves augmenting these rules to refine and to reduce both false positives and false negatives. It is a rule-based approach and considers the attributes for matches. This approach is very much dependent upon the process of evolving right rules for resolution. Relational clustering for entity resolution is proposed in [5]. Here, both attribute value and the relational information is used for associating a record with an entity. Liu, X. et al. [26] propose a method for clustering tweets which share a common topic. The Linear Conditional Random Fields model is used for labeling tweets, and then similar tweets are clustered together using these labels.

Xiao, C. et al. [39] proposed a model to join multiple records by the similarity between the corresponding values of a given attribute. The two primary methods incorporated by this approach are - 1) application of similarity function over attribute values and 2) to declare a threshold above which it should be considered as a match. A rule-based approach proposed in [17] uses the following three similarity measures to compute similarity – Winkler, Tokenset and Trigram. The similarity threshold is of two types i.e. upper and lower. Anything above the upper threshold means a proper match and below lower threshold means a non-match and between them is a possible match. The idea of quantitative similarity between attribute values is one of the fundamental techniques for entity resolution, which we have incorporated in our proposed method for entity profiling. However, we modified it to suit our model and purpose. A novel method for retrieval of relevant blog posts corresponding to a query about an entity by using a facet based information retrieval model is proposed in [37].

Benjelloun, O. et al. [3] discusses the pairwise resolution of the entities. The novelty of this approach is the usage of the match and merge properties. The four properties considered here are Idempotence, Associativity, Commutativity and Relativity. The validity of these properties, once confirmed, helps in efficient entity resolution. Bilgic, M. et al. [7] proposed a method to solve entity resolution problem with Markov logic. This method proposes combining of first-order logic and probabilistic graphical models. Weights are associated with first-order formulas and taken as features for Markov networks. The combining of first-order logic and Markov network leads to proper learning and efficient solution to the problem. A theoretical framework for knowledge-based entity resolution using first-order logic is proposed in [32]. The focus is on the analysis of knowledge patterns for optimizing a knowledge model, which is then used for entity resolution. A method for linking words or phrases in unstructured texts to entities using part of speech patterns in text is proposed in [40].

Benny, S. et al. [4] proposed a Hadoop framework for entity resolution using the Map and Reduce algorithm on big data. A technique for carrying out entity resolution on heterogeneous distributed probabilistic data is proposed in [16]. They use the expectation maximization algorithm for integrating the data. They have reported significant performance improvements over existing methods for entity resolution in a distributed environment. Ayat, N. et al. [2] proposed a method for entity resolution on probabilistic data. They have proposed algorithms for entity resolution using context-free and context-sensitive similarity functions. Hu, W. et al. [21] have proposed a scalable technique to address the problem of entity linkage on the semantic web. They have used a bootstrapping method, taking into consideration both the semantic co-referent entities as well as the similarity between property values of the entity for entity resolution.

Wang, J. et al. [38] have proposed a hybrid technique for entity resolution using both a human and a system. A system does the initial processing of data, and the users have to choose the correct pair from the most likely pairs identified by the system. Cheng, G. et al. [11] have worked on semi-automatic data integration. They have proposed a technique for selection of features to be used for interactive entity resolution, which makes use of human users to carry out the task of entity resolution. The features are selected in such a way that they convey the largest amount of diverse and characteristic information about an entity. However, this may slow down the entity profiling process, and the performance of the system will be dependent on the knowledge of the people involved. The FEVER framework [22] is used to analyze different entity resolution approaches.

3 Terms and Definitions

3.1 Similarity

Between Record and Query For finding the similarity between a record and a query, the similarity between the corresponding attributes needs to be computed. If the attributes are numeric in nature, then we use percentage difference for assigning the similarity. In the case of strings, we use the distributed representations of words and phrases [29]. A pre-trained model[2], which contains 300 dimension vectors for 3 million words and phrases is used. For this purpose, we used the Word2Vec[3] tool for computing similarity. If the words are not found in this model, we use Levenshtein distance for measuring the similarity.

[2] GoogleNews-vectors-negative300.bin.gz
[3] http://code.google.com/archive/p/word2vec

Consider finding the example of similarity between $q1$ in Table 1 and $r4$ in Table 2. The similarity for first attribute values("*Gavaskar*" and "*Gavaskar*") is 1.0. As the second attribute is $NULL$ in the query, the similarity is very small, i.e., 0.0001. For the third attribute, the values are identical which makes similarity equal to 1.0. Again the last attribute is $NULL$ in the query. Thus, the similarity value is 0.0001. By summing the similarity of all the attributes, we get 2.0002(1.0 + 0.0001 + 1.0 + 0.0001). Therefore, the similarity between the $q1$ and $r4$ is 2.0002.

Between Records For finding similarity between two records, we have used the same procedure as above. Calculating similarity between record $r6$ and $r10$ in Table 2 requires calculation of similarity for every attribute value. The first attributes ("$YuvrajSingh$" and "$YSingh$") have a similarity of 0.746. The second, third and fourth attributes have a similarity of 1. By summing the similarity of all the attributes, we get 3.746(0.746 + 1.0 + 1.0 + 1.0). Thus, the similarity between $r6$ and $r10$ is 3.746.

Between Sources The similarity between the sources is used in Section 3.6 for calculating the *trustworthiness* of a source. Let R_1 and R_2 be two sets of records published by sources S_1 and S_2 respectively. Then, the similarity between S_1 and S_2 is given by Equation 1.

$$Source\_Similarity(S_1, S_2) = \frac{\sum_{r \in R_1} \max\limits_{r' \in R_2}(similarity(r, r'))}{|R_1|} \tag{1}$$

For example, let us compute the similarity between sources $s4$ and $s1$, given in Table 2. The similarity of $r10$ with $r1$, $r2$ and $r3$ is 0.22, 0.28 and 0.46 respectively. The maximum value is 0.46. Since $s4$ has only one record, the similarity between $s4$ and $s1$ is 0.46. If $s4$ were to have more records, we will repeat this procedure for each record in $s4$, and the similarity between $s4$ and $s1$ will be sum of all the maximum values obtained divided by the number of records in $s4$.

3.2 Eligibility

Eligibility of a record with respect to a cluster is defined as a condition according to which the record can be associated with a given cluster. This association is determined at run time. The eligibility can be decided by some rules or can be determined by using a learning model. Effective identification of conditions can improve the overall accuracy of an entity profiler. Our proposed model identified most of the records related to the given query. The challenge here is to associate correct values to the missing attributes of a query from these records. The overall accuracy of a system is dependent on this step. If wrong records are selected, then in profiling phase, this will impact the selection of values to attributes.
The use of similarity threshold is discussed in [24]. However, there is no standard

procedure for selection of the eligibility criteria. The value of this threshold cannot be kept high, as there can be a case where no record will get associated. At the same time this cannot be kept low, as the wrong records may get selected for a cluster. Therefore, the threshold value selection is a non-trivial and a challenging issue in resolution phase. Due to the difficulty in the later approach, we have adopted learning model in this paper. The classifiers are trained to identify the records accurately. Given the training data, they distinguish the genuine records from the non-genuine ones.

3.3 Source rating

Source rating is a value associated with a given source, which is a quantitative measure of its trustworthiness. Since trustworthiness is relative, so are the values in the source ratings. A source S_1 (a biased source) can arbitrarily be given a value and other sources will assume source rating values depending on the similarity with S_1.
We have organized our sources into indexes. Therefore, instead of names we identify the sources by their indexes. After assigning the rating to indexes, we find the source with the maximum rating. The index of this source is termed as the index of source with maximum rating. For example, let $s1$, $s2$ and $s3$ be the sources. Consider the source $s1$ as the biased source, with the value 2.0 assigned to it. The similarity of $s1$, $s2$ and $s3$ with $s1$ is 1.0, 0.4 and 0.5 respectively. The rating of other sources will now be decided according to the source of maximum rating. Therefore, the rating for $s2$ is (2.0 / 1.0 * 0.4)=0.8 and $s3$ is (2.0 / 1.0 * 0.5)=1.0.

3.4 Attribute Value Set

After linking records to a query in resolution phase, a set $attribute\_value\_set_i$ is defined for each attribute i. The set $attribute\_value\_set_i$ will have all the values of attribute i from the selected records. We call these sets as attribute value sets. Table 5 gives us records that can be linked to q_3. We form attribute value sets for each attribute. The attribute value set $\{4,40,1,40\}$ is for the attribute "Matches". Likewise, The attribute value set $\{109,1900,10,1900\}$ is for the attribute "Runs".

3.5 Source-Similarity Matrix

The source similarity matrix gives the similarity between the sources. If we visualize every source as a vertex and edges be the similarity between the sources, then the source-similarity matrix is an adjacency matrix representation of the graph. The source-similarity matrix will be denoted as src_sim_mat in the algorithms proposed. The source similarity matrix obtained by applying Algorithm

2 on Table 2 for sources $s1$,$s2$,$s3$ and $s4$ is as follows.

$$\begin{bmatrix} - & s1 & s2 & s3 & s4 \\ s1 & 1.0000 & 0.3141 & 0.2564 & 0.1602 \\ s2 & 0.3199 & 1.0000 & 0.2868 & 0.2124 \\ s3 & 0.2803 & 0.2968 & 1.0000 & 0.1802 \\ s4 & 0.2277 & 0.4671 & 0.2380 & 1.0000 \end{bmatrix}$$

3.6 Trustworthiness

It is an indicator of the correctness and completeness of the records published by a source. The source that is more similar to other sources is considered to be the most trustworthy source as given in Equation 2. $source\_similarity(x, y)$ is the similarity between the sources x and y, as discussed in Section 3.1. Let n be the number of sources. We consider the source which is most similar to all other sources as the central or the most trustworthy source S_{MTS}, where MTS is the index of the most trustworthy source. The trustworthiness of a source S_i is computed by taking its similarity with the most trustworthy source, as given in Equation 3.

$$MTS = \underset{1 \le i \le n}{\arg\max} \left(\sum_{j=1}^{n} source\_similarity(S_i, S_j) \right) \tag{2}$$

$$Trustworthiness(S_i) = source\_similarity(S_i, S_{MTS}) \tag{3}$$

Let us consider the sources in Table 2 and the corresponding source similarity matrix given in Section 3.5. Using Equation 2, we first compute the sum of similarity of each source with all the sources. This value comes out to be 1.73, 1.82, 1.76 and 1.93 for sources $s1$, $s2$, $s3$ and $s4$ respectively. The source $s4$ is selected as the most trustworthy source (S_{MTS}). The trustworthiness of a source is its similarity with $s4$. Thus, the trustworthiness of sources $s1$, $s2$, $s3$ and $s4$ are 0.16, 0.21, 0.18 and 1.0 respectively.

3.7 Similarity-Frequency Product of an Item

Let R be the set of records associated to a query q. For an attribute a of R, let A be the set of distinct values that the records in R can take on a. We define the similarity-frequency product for each value $v \in A$ as follows. Let S denote the similarity between q_a and v, where q_a refers to the value of attribute a for a query q. Let F denote the number of records in R of attribute a having value v. Let T be the sum of similarity of v with all other values in A. The similarity-frequency product of v is computed as the product of S, F and T. It is used in Section 4.1 for selecting the correct value for an attribute from a set of possible values.

4 ReLiC: Two-Phase Entity Profiling Framework

4.1 Introduction

There are two phases defined in our model. The first phase is the resolution phase. This phase links the records referring to a query. A record is linked to a query if it is an eligible record as explained in Section 3.2. After resolution, the next phase is selection phase, where we compute the values of attributes from these records.

Resolution Phase The resolution phase associates the records with a given query. We employ a learning model for classifying the records that should be linked to a query. The performance of this phase is crucial to the overall performance of the system. A suitable classifier is selected for this phase (details in Section 5.2). Since the primary purpose of the classifier is to classify the records as relevant or non-relevant corresponding to a query, the input features should capture the information related to the record and the query. Corresponding to each attribute, we consider the similarity between the query and the value in the record, as described in Section 3.1, as an input feature for our model. A record may not have the complete information about an entity, i.e., some attribute values may be missing. To capture this in our model, for each attribute, we have an input feature which can take values 0 or 1, where 0 corresponds to a missing value. Further, we also consider the trustworthiness of the source, as described in Section 3.6, as an input feature for the classifier. Let us take the following query q and record r, where the attributes correspond to restaurant name, phone number, website and city respectively.

r=<Pizza Point, 909476941, NULL, Varanasi>
q=<Pizza Corner, 785561264, NULL, NULL>

Here, the similarity between the corresponding attributes is 0.36[4], 0.14, 0.0 and 0.0 respectively. Note that similarity for the last two attributes is 0.0 due to the missing values in record or query. These similarity values are an input feature for the classifier. Since the record has the third attribute as NULL, the feature that captures the missing information in the record, corresponding to each attribute, will take values 1, 1, 0 and 1 respectively. Let the trustworthiness of the source that published r be 0.62. In this case, the input feature to the classifier will be (0.36, 0.14, 0.0, 0.0, 1, 1, 0, 1, 0.62).

The labels are obtained for each record query pair using the ground truth data. With these input features and labels, we train the classifier using 80% of the queries and the remaining queries are used for testing the system.

Selection Phase The records linked to a query are selected and processed to complete the profile. The values used in a profile are selected from the linked

[4] Note that this value is computed using Word2Vec

records. We compute different attribute value sets as described in Section 3.4. Consider an example of $q1$ in Table 1. After entity resolution for $q1$, we get Table 3. Table 3 gives us records that can be linked to q_1. We form attribute value sets for each attribute. The attribute value set *SM Gavaskar, Gavaskar, Sunil Gavaskar* is for the attribute "Name".
In this phase, we compute similarity-frequency product as discussed in Section 3.7. For a given attribute value, it is given as the product of similarity with the query value for that attribute, it's frequency and sum of similarity with other attribute values in that attribute. For example, let us take "Name" attribute value set for query $q1$. Let us compute similarity-frequency product for attribute value *Gavaskar*. Let the similarity of *Gavaskar* with *SM Gavaskar* be 0.7 and *Sunil Gavaskar* be 0.6. The sum of similarity is 1.30. Since the query value is also *Gavaskar*, therefore similarity is 1. The frequency of *Gavaskar* in attribute value set of "Name" is 1. Therefore, its similarity-frequency product is (0.6+0.7)*1*1, i.e., 1.30. Now, two different values are obtained. First is the similarity to the source with maximum source rating and second is similarity-frequency product. The attribute value whose product of two values is maximum is chosen to be assigned to the attribute. If *Sunil Gavaskar* has the product of these two terms as maximum then it will be assigned as the value for attribute "Name". The other attribute value sets have only one element. The attribute value set for "matches" has 125, "Runs" has 10122 and "Highest" has 236. Therefore, other attributes in profile have to assume the values in their attribute value set respectively. The final profile for $q1$ is shown in Table 6.

4.2 Proposed Algorithms

Algorithm 1 gives the procedure to compute the similarity between two records. A variable *sim_value* is used to store the result. We have two values for each attribute, one by record a and the other by record b. Both these values are given as input to $similar(x, y)$ function. We compute the sum of values returned by similarity function for each attribute. The similar function is polymorphic in nature. The choice of which version to use is dependent on the type of the attribute(string or real number).
Let us compute the similarity between two records $r1$ and $r4$ in Table 2. The records have four attributes i.e. "Name", "Matches", "Runs", "Highest". The similarity between the attribute values *SM Gavaskar* and *Gavaskar* be 0.70. Rest of the attribute values(for attributes "Matches", "Runs" are "Highest") are identical. Therefore, the similarity between them will be 1.0. Thus, the similarity between the records will be 0.70 + 1.0 + 1.0 + 1.0 = 3.7.
Algorithm 2 is to compute the source similarity matrix. Our task is to form a graph, where sources are represented by vertex and similarity between them is represented by an edge between them. The similarity between the two sources is given by the sum of similarities between their records. The similarity function in Algorithm 2 takes two records as input. First record is from s_1 and second from s_2. We generate a matrix, with rows and columns representing the sources, and the values representing similarity between the two sources. Each element in the

Algorithm 1 Similarity Function

Input: Records a and b
Output: Similarity value between a and b

```
sim_value ← 0
for each attribute i do
    sim_value += similar(a_i, b_i) {Details in Section 3.1}
end for
return sim_value
```

matrix is divided by the product of the number of records in the two sources .

Algorithm 2 Source Similarity Matrix

Input: Set of sources S
Output: Source similarity matrix src_sim_mat

```
for every source s_1 do
    for every source s_2 do
        src_sim_mat_{s_1,s_2} ← 0
        for every record r_1 ∈ s_1 do
            maxsim ← 0
            for every record r_2 ∈ s_2 do
                maxsim ← max(maxsim, similarity(r_1, r_2))
            end for
            src_sim_mat_{s_1,s_2} + = maxsim
        end for
    end for
end for
for every source s_1 do
    for every source s_2 do
        src_sim_mat_{s_1,s_2} ← src_sim_mat_{s_1,s_2} / |s_1|
    end for
end for
```

Algorithm 3 is for entity profiling. Let us consider the set of records in Table 2 and query $q1$ in Table 1. In lines 1-5, we associate records to q_1 using the pre-trained classifier model as discussed in Section 4.1. After this, we get all the records linked to $q1$ as given in Table 3. Next, we assign rating to all the sources as described in Section 3.3. Let us take three sources – $s1$, $s2$ and $s3$ having indexes 1, 2 and 3 respectively. Let the ratings assigned to them be 1.0, 2.1 and 0.8 respectively. The source index with maximum rating is assigned to the $index\_of\_maximum$. The $index\_of\_maximum$ in our case is 2, corresponding to $s2$. We generate attribute value set for each attribute, containing all distinct values that the attribute assumes. The attribute value set for the attribute "Name" is *SM Gavaskar, Gavaskar, Sunil Gavaskar*, for "Matches" is $\{125\}$, for "Runs" is $\{10122\}$ and for "Highest" is $\{236\}$. In lines 20-24, we calculate the sum of similarity of an attribute value with other values of that attribute. This sum, for a given attribute value v and attribute i, is denoted by $sim\_attribute\_val_{v,i}$. We assume that majority of the values in attribute value set have high similarity with each other, and the correct value resides among them. To find this value, we take the sum of similarity of an attribute value with all other values. This way, the majority of the values which are similar to each other get high $sim\_attribute\_val_{v,i}$. Further, the attribute value which is central, i.e., most similar to all other values gets the highest $sim\_attribute\_val_{v,i}$. We use $sim\_attribute\_val$ in the calculation of the similarity-frequency product. Further, for every value in attribute value set for attribute i, we calculate its similarity-frequency product as described in Sections 3.7 and 4.1. Finally, the attribute value whose product of the similarity to $index\_of\_maximum$ and the similarity-frequency product is maximum is chosen for assignment.

5 Experimental Setup and Results

In this section, we present the results of the extensive experiments done to validate the proposed methodology.

5.1 Datasets

For our experiments, use real world datasets on restaurant, football and cricket. All these datasets have records extracted from multiple sources.

Restaurant Dataset The restaurant dataset in [24] contains information about restaurants with the zip code 78701. It consists of 581 records on 222 restaurants, extracted from 6 websites. Each record has values corresponding to name, address, phone, website and id. The ground truth is obtained from extracting information about these restaurants from Yellow Pages[5]. Each query comprises of two missing attributes and three filled attributes.

Football Dataset The football dataset in [8] contains information about football players. It consists of 7492 records extracted from 20 websites. Each record has values corresponding to name, birth date, height, weight, playing position and birth place. The ground truth were extracted from the official sites of the football clubs which these players represent. In [24], a series of datasets are generated using the original football dataset by introducing errors and ambiguities in the data. The errors are generated by randomly generating an erroneous value and ambiguities are generated by abbreviating names, removing first name or removing the last name. The percentage error is varied from 0.1-0.5, whereas

[5] www.yellowpages.com

the percentage ambiguity is varied from 0.4-0.9. Each query comprises of three missing attributes and four filled attributes.

Algorithm 3 Entity profiling

Input: Query q, set of records R, set of sources S
Output: Completed profile p

for every $r \in R$ **do**
  $s \leftarrow source(r)$
  if classifier.predict(r,q,s) == 1 **then**
    {Classifier predicts r as genuine with respect to q}
    associate r with q
  end if
end for
$src\_rating \leftarrow$ list of ratings of all sources
$index\_of\_maximum \leftarrow max\_index(src\_rating)$
for every attribute i **do**
  $attribute\_value\_set_i \leftarrow \phi$
  for every record r associated to q **do**
    $attribute\_value\_set_i \leftarrow attribute\_value\_set_i \cup r_i$
  end for
end for
for every attribute i **do**
  for every value $\in attribute\_value\_set_i$ **do**
    Initialize dictionary $sim\_attribute\_val_{value} \leftarrow 0$
  end for
  for every $value_1 \in attribute\_value\_set_i$ **do**
    for every $value_2 \in attribute\_value\_set_i$ and $value_1 \neq value_2$ **do**
      $sim\_attribute\_val_{value_1}$ += *similar($value_1$, $value_2$)*
    end for
  end for
  for every value v in $attribute\_value\_set_i$ **do**
    $V1_{v,i} \leftarrow src\_sim\_mat_{index\_of\_maximum,s}$ {s is the source from which the v comes}
    $V2_{v,i} \leftarrow similarity\_frequency\_product_{v,i}$ {Explained in Section 3.7}
    $V3_{v,i} \leftarrow V1_{v,i} * V2_{v,i}$
  end for
  The value in $attribute\_value\_set_i$ having the maximum $V3_{v,i}$ is assigned to p_i
end for

Cricket Dataset We prepared cricket dataset by extracting statistical information on cricketers in Test Cricket. The data comprises of 330 records extracted from 3 websites. Each record has values corresponding to name,the number of matches, net runs, highest score, batting average, number of hundreds, number

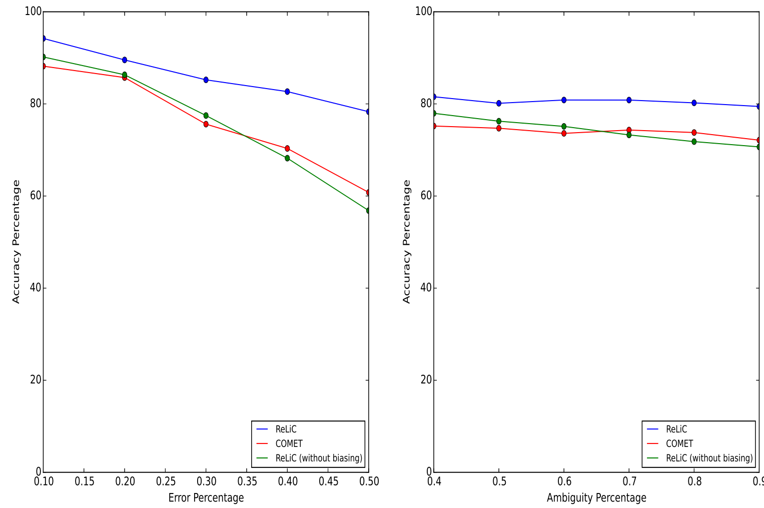

Fig. 1. Accuracy for Football Dataset

of fifties and country. The ground truth is obtained from the Wikipedia pages of each of these players. Each query comprises of four missing attributes and four filled attributes.

5.2 Selection of Classifier Model

As discussed in Section 4.1, an appropriate classifier model needs to be selected for assigning records to the queries. For this purpose, we randomly select 80% of the records as training data and remaining as testing data. We calculate the $F1$ score, cross correlation error, ROC AUC score [9] and the Matthews Correlation Coefficient (MCC) [28] for different classifiers on the testing data. The $F1$ score is the harmonic mean of the precision and recall. The cross-validation error was computed using 10 fold strategy; wherein the data is divided into 10 subsets of equal length. During each iteration, one of the subsets is considered as a test set and the rest are used for training. ROC AUC score refers to the area under the receiver operating characteristic (ROC) curve. The ROC curve is a plot between the recall and the fall-out of the classifier. It is a standard technique used for the comparison of classifier models [20]. The Matthews Correlation Score (MCC) is used for measuring the quality of binary classifiers. TP, FP, TN and FN denotes true positives, false positives, true negatives and false negatives respectively. MCC is computed using Equation 4.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4)$$

The results of various classifier models is given in Table 7. We used scikit-learn [30] python library for implementing these classifiers. Note that, we used SVM with linear kernel, and random forest with 10 trees. From the Table 7, we can conclude that the Random Forest Classifier clearly outperforms all the other classifiers. Therefore, we have chosen this classifier for obtaining the subsequent results.

Table 7. Performance of Classifier Models

Classifier Model	**F1 Score**	**Cross Validation Error (%)**	**ROC AUC Score**	**MCC**
Naive Bayes	0.8412	6.23	0.8494	0.7295
K Nearest Neighbour	0.8724	4.24	0.9177	0.8362
SVM	0.9088	4.33	0.9315	0.8344
Decision Tree	0.8973	4.15	0.9035	0.8011
Random Forest	0.9567	2.72	0.9511	0.9124

5.3 Results

We compare ReLiC with the COMET framework [24], which is the state-of-the-art entity profiling technique. We evaluate our model for entity resolution and also for overall profile completion. Results are evaluated using three metrics - accuracy, precision and recall.

Precision and recall are used for evaluating the entity resolution phase. They are calculated on the records associated with each query using Equations 5 and 6. Let R_i be the set of records associated with query q_i and R'_i be the set of records that get associated to query q_i during the resolution phase of the algorithm.

$$Precision = \frac{|R_i \cap R'_i|}{|R'_i|} \quad (5)$$

$$Recall = \frac{|R_i \cap R'_i|}{|R_i|} \quad (6)$$

The accuracy metric is used for evaluating the entire entity profiling approach. It refers to the amount of similarity between the query filled by the entity profiling approach and the ground truth values corresponding to that query, as seen in

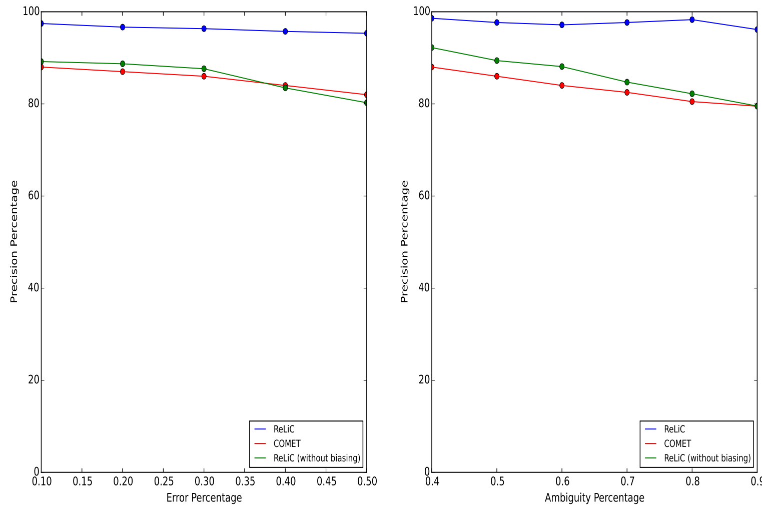

Fig. 2. Precision for Football Dataset

Equations 7 and 8. Here, similarity between the query and ground truth values is calculated by taking the average of the similarity between the corresponding attributes. If the attributes are numerals, then we use the ratio of difference, otherwise, we use Levenshtein distance [23] for computing the similarity between two attributes.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|A|}\sum_{a\in A} sim(q_a, t_a) \tag{7}$$

Here, N denotes the number of filled queries, q denotes the filled query, t denotes the corresponding truth values and A denotes the set of all attributes.

$$sim(v_1, v_2) = \begin{cases} \frac{|v_1 - v_2|}{max(v_1, v_2)} & \text{if } v_1, v_2 \in \mathbb{R} \\ Levenshtein(v_1, v_2) & \text{otherwise} \end{cases} \tag{8}$$

Here, $Levenshtein(a, b)$ refers to the edit distance between a and b.

We have also calculated the accuracy, precision and recall values for ReLiC framework, without the use of biased sources. In this case, we have kept the rating of all the sources to be equal. It is evident that there is a significant impact on the performance of the system with biasing. From this, we can infer that biasing boosts the performance of a profiling system.

For the ReLiC framework, we use 70% of the queries for training the model, and the remaining queries are used for testing. We calculate the accuracy, precision and recall to test the performance of all the models. Tables 8 and 9 show

Table 8. Results for Restaurant dataset

Methodology	Accuracy	Precision	Recall
ReLiC	**93.7913**	**97.1965**	**96.2475**
COMET	86.2474	95.2671	95.1111
ReLiC (without biasing)	82.6437	93.6403	92.4048

Table 9. Results for Cricket dataset

Methodology	Accuracy	Precision	Recall
ReLiC	**87.2733**	**92.9083**	**92.0222**
COMET	80.9424	88.6667	86.2471
ReLiC (without biasing)	82.2431	88.3534	87.7356

the results for the restaurant and cricket datasets respectively. Figures 1, 2 and 3 show the performance of all the systems on football dataset. From these results, we can conclude that our proposed system outperforms the current state-of-the-art entity profiling framework.

Table 10. Result for 2-sided paired student's t-test on ReLiC and COMET for Restaurant dataset

Metric	T Value	P Value	Sample Effect Size
Precision	2.9600	0.00473	0.9239
Recall	2.8189	0.00693	1.0911
Accuracy	3.5213	0.00094	1.3158

For validating the statistical significance of our experiments, we have conducted a 2-sided paired student's t-test as described in [31]. For this purpose, we have compared ReLiC with COMET, using all the 3 datasets (cricket, football, restaurant) on each of the performance metrics – precision, recall and accuracy. We calculate the T value, P value and the sample effect size. The value of α is chosen as 0.05. The values are reported in Tables 10, 11 and 12. From these tables, it is clear that the obtained P values are less than α. Further, T values and sample effect sizes are considerable. This shows that ReLiC significantly outperforms COMET.

5.4 Discussion

Various knowledge bases are entirely devoted to the collection of data related to their fields. Owners, organisations or followers regularly update these sources.

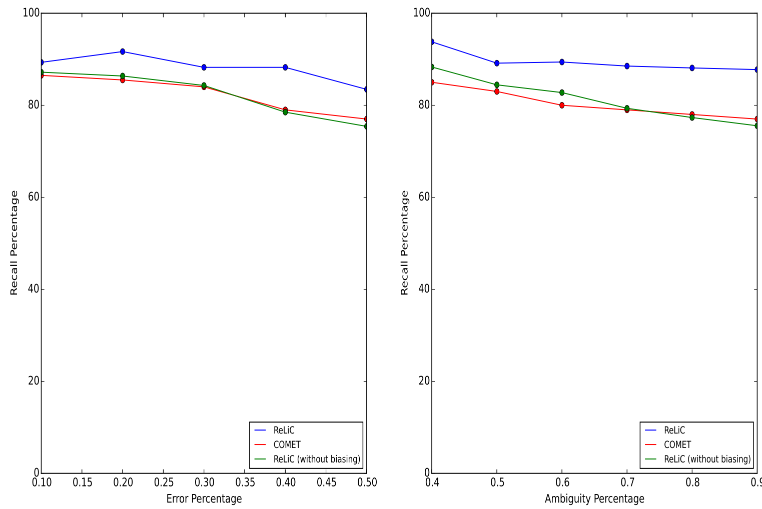

Fig. 3. Recall for Football Dataset

Table 11. Result for 2-sided paired student's t-test on ReLiC and COMET for Football dataset

Metric	T Value	P Value	Sample Effect Size
Precision	2.5102	0.01542	0.9420
Recall	3.3164	0.00172	1.2563
Accuracy	2.3313	0.02390	1.3589

Table 12. Result for 2-sided paired student's t-test on ReLiC and COMET for Cricket dataset

Metric	T Value	P Value	Sample Effect Size
Precision	3.8382	0.00036	1.0762
Recall	3.0846	0.00334	1.2757
Accuracy	2.5009	0.01578	1.2444

Thus, the information content in these sources is generally accurate. To give more importance to the data published by these sources, we have used the concept of a biased source for entity profiling. The concept of biased source allows such sources to have higher ratings as compared to others. The results demonstrate that the use of biased sources is effective for entity profiling. As seen in Table 8, the use of biased source in ReLiC has shown an increase of around 11% in accuracy and around 4% in precision and recall.

COMET is the state-of-the-art framework for entity profiling. It uses a clustering-based technique for associating records to queries. Further, it uses pruning for removal of irrelevant records from the queries. On the other hand, ReLiC uses the random forest for associating records to queries. Table 7 shows that random forest classifier has a high F1 score of 0.9567. Since random forest links most of the relevant records to the queries, pruning is not required. The next step involves selection of value from the set of attribute values obtained. COMET uses source reliability for this purpose, whereas ReLiC uses the similarity-frequency product for selecting the correct attribute value.

6 Conclusions

In this paper, we address the problem of entity profiling using a two-phase framework (ReLiC). In the first phase, we assign ratings to the sources by computing their similarity with respect to a biased source. We have used Word2Vec for computing similarity between strings. We associate records to a query using random forest classifier. Our results demonstrate that the usage of a biased source improved the overall performance of ReLiC. We used three similarity measures - query-record, record-record and source-source to address the problem of entity profiling. We compared our system with the state-of-the-art COMET framework and our results clearly demonstrate that ReLiC outperforms COMET. ReLiC framework can be further improved by considering temporal information associated to the records. This can allow the user to get the complete profile of an entity at different points of time. Further, ReLiC and COMET assume that the details given in the user query are accurate. Therefore, considering errors in user query may improve the performance of ReLiC.

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, pages 722–735. Springer Berlin Heidelberg, 2007.
2. Naser Ayat, Reza Akbarinia, Hamideh Afsarmanesh, and Patrick Valduriez. Entity resolution for probabilistic data. *Information Sciences*, 277:492 – 511, 2014.
3. Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: A generic approach to entity resolution. *The VLDB Journal*, 18(1):255–276, 2009.

4. S. Prabhakar Benny, S. Vasavi, and P. Anupriya. Hadoop framework for entity resolution within high velocity streams. *Procedia Computer Science*, 85:550 – 557, 2016. International Conference on Computational Modelling and Security (CMS 2016).
5. Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1):1–36, 2007.
6. Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48. ACM, 2003.
7. M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman. D-dupe: An interactive tool for entity resolution in social networks. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 43–50, 2006.
8. Lorenzo Blanco, Valter Crescenzi, Paolo Merialdo, and Paolo Papotti. *Web Data Reconciliation: Models and Experiences*, pages 1–15. Springer Berlin Heidelberg, 2012.
9. Andrew P. Bradley. The use of the area under the {ROC} curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145 – 1159, 1997.
10. Silvia Calegari and Gabriella Pasi. Personal ontologies: Generation of user profiles based on the {YAGO} ontology. *Information Processing & Management*, 49(3):640 – 658, 2013. Personalization and Recommendation in Information Access.
11. Gong Cheng, Danyun Xu, and Yuzhong Qu. C3d+p: A summarization method for interactive entity resolution. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35, Part 4:203 – 213, 2015.
12. Jerome Cheng, Kazunari Sugiyama, and Min-Yen Kan. Linking organizational social network profiles. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 901–904, New York, NY, USA, 2016. ACM.
13. P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Berlin Heidelberg, 2012.
14. Peter Christen. Febrl: A freely available record linkage system with a graphical user interface. In *Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management - Volume 80*, pages 17–25. Australian Computer Society, Inc., 2008.
15. Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphal Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32 – 49, 2015.
16. Ramesh Dharavath and Chiranjeev Kumar. Entity resolution based {EM} for integrating heterogeneous distributed probabilistic data. *Journal of Systems and Software*, 107:93 – 109, 2015.
17. Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
18. Anja Gruenheid, Xin Luna Dong, and Divesh Srivastava. Incremental record linkage. *Proc. VLDB Endow.*, 7(9):697–708, 2014.
19. Songtao Guo, Xin Luna Dong, Divesh Srivastava, and Remi Zajac. Record linkage with uniqueness constraints and erroneous values. *Proc. VLDB Endow.*, 3(1-2):417–428, 2010.
20. J A Hanley and B J McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.

21. Wei Hu and Cunxin Jia. A bootstrapping approach to entity linkage on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 34:1 – 12, 2015.
22. Hanna Köpcke, Andreas Thor, and Erhard Rahm. Comparative evaluation of entity resolution approaches with fever. *Proc. VLDB Endow.*, 2(2):1574–1577, 2009.
23. Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710, 1966.
24. Furong Li, Mong Li Lee, and Wynne Hsu. Entity profiling with varying source reliabilities. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1146–1155. ACM, 2014.
25. Furong Li, Mong Li Lee, Wynne Hsu, and Wang-Chiew Tan. Linking temporal records for profiling entities. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 593–605. ACM, 2015.
26. Xiaohua Liu and Ming Zhou. Two-stage {NER} for tweets with clustering. *Information Processing & Management*, 49(1):264 – 273, 2013.
27. Bradley Malin and Latanya Sweeney. Enres: a semantic framework for entity resolution modelling. In *Institute for Software Research International Technical Report*. Carnegie Mellon University, 2005.
28. B.W. Matthews. Comparison of the predicted and observed secondary structure of {T4} phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442 – 451, 1975.
29. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.
30. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.
31. Tetsuya Sakai. Statistical reform in information retrieval? *SIGIR Forum*, 48(1):3–12, June 2014.
32. Klaus-Dieter Schewe and Qing Wang. A theoretical framework for knowledge-based entity resolution. *Theoretical Computer Science*, 549:101 – 126, 2014.
33. P. Singla and P. Domingos. Entity resolution with markov logic. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 572–582, 2006.
34. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706. ACM, 2007.
35. Huzaifa Syed, John Talburt, Fan Liu, Daniel Pullen, and Ningning Wu. Developing and refining matching rules for entity resolution. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, pages 1–6, 2012.
36. J.R. Talburt. *Entity Resolution and Information Quality*. Elsevier Science, 2011.
37. Olga Vechtomova. Facet-based opinion retrieval from blogs. *Information Processing & Management*, 46(1):71 – 88, 2010.
38. Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, 2012.
39. Chuan Xiao, Wei Wang, Xuemin Lin, and Jeffrey Xu Yu. Efficient similarity joins for near duplicate detection. In *Proceedings of the 17th International Conference on World Wide Web*, pages 131–140. ACM, 2008.

40. Lei Zhang, Yunpeng Dong, and Achim Rettinger. Towards entity correctness, completeness and emergence for entity recognition. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 143–144, New York, NY, USA, 2015. ACM.
41. Gang Zhao, Ji Wu, Dingding Wang, and Tao Li. Entity disambiguation to wikipedia using collective ranking. *Information Processing & Management*, 2016. In Press.