ReLiC: entity profiling using random forest and trustworthiness of a source

SHUBHAM VARMA¹, NEYSHITH SAMEER¹ and C RAVINDRANATH CHOWDARY^{2,*}

¹Microsoft Research and Development, Hyderabad, India

²Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India
e-mail: shvarma@microsoft.com; nesameer@microsoft.com; rchowdary.cse@iitbhu.ac.in

MS received 26 February 2018; revised 8 May 2019; accepted 2 July 2019

Abstract. The digital revolution has brought most of the data in the world to the world wide web, but at the same time, the data available on WWW has increased manyfold in the past decade. Social networks, online clubs etc., have come into existence. Expert systems are required to extract information 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 to maintain the data. Therefore, it is desirable to have an intelligent 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 studied the performance of various classifiers for this purpose and observed that the use of the random forest is best suited for entity resolution. Also, we proposed and 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 used the concept of assigning authority to a reliable source in entity profiling, and it is established through our results that the use of an authoritative source has significantly improved the performance of the proposed system. Experimental results show that our proposed system ReLiC outperforms COMET.

Keywords. Entity profiling; entity resolution; record linkage; authoritative source.

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 an intelligent 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 these 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 essential step for entity profiling. In this paper, we study the use of classifiers for entity

resolution. Due to this, the proposed method worked effectively on different types of data. Entity profiling helps in acquiring accurate information from data extracted from various unreliable sources. However, to the best of our knowledge, assigning authority to a reliable source for profiling was not considered earlier in the literature and this motivated us to use an authoritative 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 its 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

*For correspondence
Published online: 20 August 2019

Table 1. Table of queries.

Query	Cluster	Name	Matches	Runs	Highest
q_1	c_1	Gavaskar		10122	
q_2	c_2	Amarnath	69		
q_3	c_3	Y Singh			169

Table 2. Table of records from various sources.

Record	Name	Matches	Runs	Highest	Source
r_1	SM Gavaskar	125	10122	236	s1
r_2	Lala Amarnath	24	878	118	s1
r_3	Yajurvindra Singh	4	109	43	s1
r_4	Gavaskar	125	10122	236	s2
r_5	Mohinder Amarnath	69	4378	138	s2
r_6	Yuvraj Singh	40	1900	169	s2
r_7	Sunil Gavaskar	125	10122	236	s3
r_8	Surinder Amarnath	10	550	124	s3
r_9	Yograj Singh	1	10	6	s3
r_{10}	Y Singh	40	1900	169	s4

repositories on real-world entities. Knowledge bases such as Instant Checkmate¹ provide complete information about a person using online profiles. DBPedia [1] and YAGO [2] are other knowledge bases that provide public databases on real-world entities.

Entity profiles are represented by a set of attribute values. Let us consider table 1, which contains three queries. These queries are incomplete profiles and have to be completed using table 2. Each data item published by a source is referred to as a record and is represented using a set of attribute values. Table 2 contains records along with their respective sources. The first step in completing a profile involves associating relevant records to a query. This is known as entity resolution and is generally done by clustering similar records to the query. This creates a cluster of relevant records corresponding to each query. Every query is given a cluster-id as shown in table 1.

Let us take the first query, i.e., q_1 having its associated cluster as c_1 ; q_1 refers to the records given in table 3. Hence, we associated all the records in tables 3, 4 and 5 to the clusters c_1 , c_2 and c_3 , respectively. From these records, we complete the profile for q_1 , and the completed profile is given in table 6. Similarly, after applying entity resolution to q_3 , we get table 5. In table 5 we see that query can refer to only r_6 and r_{10} based on the attribute “Highest”. These two records are now used to complete the profile. Table 6

Table 3. Table of cluster c_1 .

Record	Name	Matches	Runs	Highest	Source
r_1	SM Gavaskar	125	10122	236	s1
r_4	Gavaskar	125	10122	236	s2
r_7	Sunil Gavaskar	125	10122	236	s3

Table 4. Table of cluster c_2 .

Record	Name	Matches	Runs	Highest	Source
r_2	Lala Amarnath	24	878	118	s1
r_5	Mohinder Amarnath	69	4378	138	s2
r_8	Surinder Amarnath	10	550	124	s3

gives completed profiles. A similar assignment holds for query q_2 .

Entity resolution is one of the critical 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 4.1a. Section 5.4 describes the technique used for the 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 concerning an authoritative source. The user selects this authoritative 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 authoritative 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

¹<https://www.instantcheckmate.com/>.

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

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

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 studied the performance of various classifiers for associating records to a query (entity resolution) and found that random forest outperforms other classifiers.
2. We have proposed the use of an authoritative source for the 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 active research problem. Li *et al* [3] 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 [4], which tries to model the evolution of an entity over time. A system for profiling entities in a ubiquitous learning environment using semantic interoperability between the various applications is proposed in [5]. The work in [6] leverages data mining techniques for mining interests of users from various chat and discussion forums for user profiling. Our technique

focuses on associating records to entities, where the records have already been mined from different data sources.

Hawalah and Fasli [7] propose a technique for modelling contextual user profiles based on the user’s interests and context information obtained from various sources. They have developed a context-aware personalization system, which models and builds such contextual profiles for providing personalized recommendations to the users. In [8], 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. This technique might not scale to different types of data since uniqueness constraints will differ in different datasets. On the other hand, our technique makes use of a supervised model that learns from the training data and thus can perform well with different kinds of data.

An ontology-based approach for the generation of user profiles using YAGO [2] is proposed in [9]. Garcia *et al* [10] have proposed an ontology-based approach for collecting and integrating analytics data obtained from e-commerce sites for profiling user behaviour. Amini *et al* [11] proposed a technique for scholar profiling by collecting and unifying the data items obtained through scholar profiles from various digital libraries.

The various models in the literature addressing 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 [12]. FEBRL [13] (Freely Extensible Biomedical Record Linkage) uses SVM to learn the appropriate matching combinations. The similarity measures used for this approach are the same as those in the rule-based approach proposed in [14]. MARLIN [15] (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 [16]. Zhao *et al* [17] propose a novel model for linking entities mentioned in texts to semi-structured knowledge bases, by considering relationships between entities that occur together more frequently in the text. This technique has been shown to disambiguate entities effectively.

Christen *et al* have extensively studied the use of various supervised [18–20] and non-supervised models [21] for record linkage. They have prominently used temporal information and record similarities as features for these models. In this paper, we have also proposed a supervised technique for record linkage considering record similarities. However, we have also considered source trustworthiness as an important factor, since different sources will very likely have different reliabilities.

Cheng *et al* [22] have used entity resolution for linking social networking profiles to the organisations (affiliations) of the respective users. An analysis of information extraction techniques for linking tweets to entities is reported in

[23]. Detailed description and analysis of various entity resolution and data linkage approaches are discussed in [24, 25].

An entity resolution model that focuses on an incremental approach to building clusters is presented in [26]. 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 *et al* [27] 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 [28]. Here, both attribute value and the relational information are used for associating a record with an entity.

Xiao *et al* [29] 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) declaring a threshold above which it should be considered as a match. A rule-based approach proposed in [14] 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.

Benjelloun *et al* [30] discuss 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 *et al* [31] proposed a method to solve the entity resolution problem with Markov logic. This method proposes combining 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 the text is proposed in [33]. On the other hand, our technique focuses on associating structured records to entities for profiling.

Benny *et al* [34] 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 [35]. 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 *et al* [36] 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 and Jia [37] 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. Similarly, we have also taken the similarity between attributes into consideration for entity profiling.

Extensive research on entity resolution, matching and profiling has been done by Yerva in [38]. A generic framework is proposed for defining multiple similarity functions corresponding to different domain-specific rules. These similarity functions are used for capturing the equivalence between entities. Further, results from these similarity functions are combined using graph theory and machine learning techniques for enhanced entity resolution. The thesis also talks about entity resolution in unstructured datasets. Various techniques have been proposed for extracting information about entities from online social network data using content mining and modelling. Our paper, on the other hand, discusses the problem of entity resolution and profiling from structured data using supervised learning techniques.

Wang *et al* [39] 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 *et al* [40] 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 [41] is used to analyse different entity resolution approaches.

3. Terms and definitions

3.1 Similarity

3.1a 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, then we use the percentage difference for assigning the similarity. In the case of strings, we use the distributed representations of words and phrases [42]. A pre-trained model² that contains 300-dimension vectors for 3 million words and phrases is used. For this purpose, we used the Word2Vec³ tool for computing similarity. If the words are not found in this model, we use Levenshtein distance for measuring the similarity.

For example, consider finding 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 $q1$ and $r4$ is 2.0002.

3.1b Between records: For finding similarity between records, we have used the same procedure described earlier. Calculating similarity between records $r6$ 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.

3.1c 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 Eq. (1):

$$\text{source_similarity}(S_1, S_2) = \frac{\sum_{r \in R_1} \max_{r' \in R_2} (\text{similarity}(r, r'))}{|R_1|}. \quad (1)$$

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 concerning a query is defined as a condition according to which the record can be associated with a given query. This association is determined at run time. The eligibility can be decided by some rules or can be determined 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 the profiling phase, this will impact the selection of values to attributes.

The use of similarity threshold is discussed in [3]. 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 wrong records may get selected for a query. Therefore, the threshold value selection is a non-trivial and a challenging issue in the resolution phase. Due to the difficulty in the latter approach, we have adopted the 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 that is a quantitative measure of its trustworthiness. Since trustworthiness is relative, so are the values in the source ratings. A source S_1 (an authoritative source) can be arbitrarily 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 authoritative 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 $((2.0 / 1.0) \times 0.4) = 0.8$ and $s3$ is $((2.0 / 1.0) \times 0.5) = 1.0$.

3.4 Attribute value set

After linking records to a query in resolution phase, a set $\text{attribute_value_set}_i$ is defined for each attribute i . The set

²GoogleNews-vectors-negative300.bin.gz.

³<http://code.google.com/archive/p/word2vec>.

$attribute_value_set_i$ 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 $s1$ – $s4$ is as follows:

—	$s1$	$s2$	$s3$	$s4$
$s1$	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

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 Eq. (2); $source_similarity(x, y)$ is the similarity between the sources x and y , as discussed in section 3.1. Here, n is the number of sources. We consider the source that is the 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 Eq. (3):

$$MTS = \arg \max_{1 \leq i \leq n} \left(\sum_{j=1}^n source_similarity(S_i, S_j) \right), \quad (2)$$

$$trustworthiness(S_i) = source_similarity(S_i, S_{MTS}). \quad (3)$$

Let us consider the sources in table 2 and the corresponding source-similarity matrix given in section 3.5. Using Eq. (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 values 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 4.1b for selecting the correct value for an attribute from a set of possible values.

4. ReLiC: two-phase entity profiling framework

4.1 Introduction

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 intelligently assign the values of attributes from these records.

4.1a 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 5.4). 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 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 that 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:

$$r = \langle PizzaPoint, 909476941, NULL, Varanasi \rangle, \\ q = \langle PizzaCorner, 785561264, NULL, NULL \rangle.$$

Here, the similarity between the corresponding attributes is 0.36⁴, 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

⁴Note that this value is computed using Word2Vec.

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.

4.1b Selection phase: The records linked to a query are intelligently selected and processed to complete the profile. The values used in a profile are selected from the linked records. We compute different attribute value sets as described in section 3.4. Consider an example of q_1 in table 1. After entity resolution for q_1 , we get records as given in 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, its frequency and sum of similarity with other attribute values in that attribute. For example, let us take “Name” attribute value set for query q_1 . 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*, similarity is 1. The frequency of *Gavaskar* in attribute value set of “Name” is 1. Therefore, its similarity–frequency product is $(0.6 + 0.7) \times 1 \times 1$, i.e., 1.30.

Now, two different values are obtained. First is the similarity to the source with maximum source rating and second is the 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. The final profile for q_1 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 of these values are given as input to *similar*(x, y) function. We compute the sum of values returned by the similarity function for each

attribute. The similarity function is polymorphic. 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 r_1 and r_4 in table 2. The records have four attributes, i.e., “Name”, “Matches”, “Runs” and “Highest”. Let the similarity between the attribute values *SM Gavaskar* and *Gavaskar* be 0.70. Rest of the attribute values (for attributes “Matches”, “Runs” and “Highest”) are identical. Therefore, the similarity between them will be 1.0. Thus, the similarity between the records is $0.70 + 1.0 + 1.0 + 1.0 = 3.7$.

Algorithm 1 Similarity Function

Input: Records a and b

Output: Similarity value between a and b

```

1: sim_value  $\leftarrow$  0
2: for each attribute  $i$  do
3:   sim_value += similar( $a_i, b_i$ ) {Details in Section 3.1}
4: end for
5: return sim_value

```

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 1 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 the similarity between the two sources. Each element in the 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*

```

1: for every source  $s_1$  do
2:   for every source  $s_2$  do
3:     src_sim_mat $_{s_1, s_2} \leftarrow$  0
4:     for every record  $r_1 \in s_1$  do
5:       maxsim  $\leftarrow$  0
6:       for every record  $r_2 \in s_2$  do
7:         maxsim  $\leftarrow$  max(maxsim, similarity( $r_1, r_2$ ))
8:       end for
9:       src_sim_mat $_{s_1, s_2} + =$  maxsim
10:    end for
11:  end for
12: end for
13: for every source  $s_1$  do
14:   for every source  $s_2$  do
15:     src_sim_mat $_{s_1, s_2} \leftarrow \frac{\text{src\_sim\_mat}_{s_1, s_2}}{|s_1|}$ 
16:   end for
17: end for

```

Algorithm 3 is for entity profiling. Let us consider the set of records in table 2 and query q_1 in table 1. In lines 1–5, we associate records to q_1 using the pre-trained classifier model as discussed in section 4.1a. After this, we get all the records linked to q_1 as given in table 3. Next, we assign rating to all the sources as described in section 3.3. Let us take three sources— s_1 , s_2 and s_3 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 *index_of_maximum*. The value for *index_of_maximum* in our case is 2, corresponding to s_2 . 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 the majority of the values in the 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 that are similar to each other get high *sim_attribute_val_{v,i}*. Further, the attribute value that 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 4.1b. Finally, the attribute value whose product of the similarity to *index_of_maximum* and the similarity–frequency product is maximum is chosen for the assignment.

5. Experimental set-up and results

In this section, we present the results of the extensive experiments done to validate the proposed methodology. For our experiments, we use real world datasets on restaurant, football and cricket. All these datasets have records extracted from multiple sources.

5.1 Restaurant dataset

The restaurant dataset in [3] contains information about restaurants with the zip code 78701. It consists of 581 records on 222 restaurants, extracted from six 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 two missing attributes and three filled attributes.

Algorithm 3 Entity profiling

Input: Query q , set of records R , set of sources S

Output: Completed profile p

```

1: for every  $r \in R$  do
2:    $s \leftarrow source(r)$ 
3:   if classifier.predict( $r, q, s$ ) == 1 then
4:     {Classifier predicts  $r$  as genuine with respect to  $q$ }
5:     associate  $r$  with  $q$ 
6:   end if
7: end for
8:  $src\_rating \leftarrow$  list of ratings of all sources
9:  $index\_of\_maximum \leftarrow max\_index(src\_rating)$ 
10: for every attribute  $i$  do
11:    $attribute\_value\_set_i \leftarrow \emptyset$ 
12:   for every record  $r$  associated to  $q$  do
13:      $attribute\_value\_set_i \leftarrow attribute\_value\_set_i \cup r_i$ 
14:   end for
15: end for
16: for every attribute  $i$  do
17:   for every value  $\in attribute\_value\_set_i$  do
18:     Initialize dictionary  $sim\_attribute\_val_{value} \leftarrow 0$ 
19:   end for
20:   for every  $value_1 \in attribute\_value\_set_i$  do
21:     for every  $value_2 \in attribute\_value\_set_i$  and  $value_1 \neq value_2$  do
22:        $sim\_attribute\_val_{value_1} += similar(value_1, value_2)$ 
23:     end for
24:   end for
25:   for every value  $v$  in  $attribute\_value\_set_i$  do
26:      $V1_{v,i} \leftarrow src\_sim\_mat_{index\_of\_maximum, s}$  { $s$  is the source from which the  $v$  comes}
27:      $V2_{v,i} \leftarrow similarity\_frequency\_product_{v,i}$  {Explained in Section 3.7}
28:      $V3_{v,i} \leftarrow V1_{v,i} * V2_{v,i}$ 
29:   end for
30:   The value in  $attribute\_value\_set_i$  having the maximum  $V3_{v,i}$  is assigned to  $p_i$ 
31: end for

```

5.2 Football dataset

The football dataset in [43] 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 birthplace. The ground truth was extracted from the official sites of the football clubs that these players represent. In [3], 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

⁵www.yellowpages.com.

and ambiguities are generated by abbreviating names, removing the first name or removing the last name. The percentage error is varied from 0.1 to 0.5, whereas the percentage of ambiguity is varied from 0.4 to 0.9. Each query comprises three missing attributes and four filled attributes.

5.3 Cricket dataset

We prepared cricket dataset⁶ by extracting statistical information on cricketers in Test Cricket. The data comprise 330 records extracted from three websites. Each record has values corresponding to name, number of matches, net runs, highest score, batting average, number of hundreds, number of fifties and country. The ground truth is obtained from the Wikipedia pages of each of these players. Each query comprises four missing attributes and four filled attributes.

5.4 Selection of classifier model

As discussed in section 4.1a, 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, receiver operating characteristics (ROC) AUC score [44] and the Matthews Correlation Coefficient (MCC) [45] for different classifiers on the testing data. The $F1$ score is the harmonic mean of precision and recall. The cross-validation error is computed using a 10-fold strategy wherein the data is divided into 10 subsets of equal length. During each iteration, one of the subsets is considered as the test set and the rest are used for training.

ROC AUC Score refers to the area under the 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 [46]. The MCC is used for measuring the quality of binary classifiers. TP , FP , TN and FN denote true positives, false positives, true negatives and false negatives, respectively. MCC is computed using Eq (4):

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

The classifier model comparison can be seen in table 7. For the implementation of these classifiers, we use the scikit-learn [47] python library. Note that the SVM classifier was with a linear kernel, and the random forest classifier was used with 10 trees. As seen in table 7, the random forest classifier proves to be the best. Due to this reason, this

Table 7. Comparing 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
KNN	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

classifier best fits the given task. Hence, we have used this classifier for obtaining the subsequent results.

5.5 Results

We compare ReLiC to the COMET framework [3], which is the state-of-the-art entity profiling technique. We also considered the entity resolution technique proposed by Hu *et al* [19], which makes use of temporal information and supervised learning techniques. A drawback is that it requires datasets to have temporal information. Since our datasets do not contain temporal information, we have not used it for our results.

We evaluate our model for entity resolution as well as overall profile completion. We compare 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 Eqs. (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. Then

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

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

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 given in Eqs. (7) and (8). Here, the 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 for calculating similarity; otherwise, we use Levenshtein distance [48] for computing the similarity between two attributes. 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:

⁶<https://github.com/ShubhamVarma/ReLiC>.

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 authoritative source)	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 authoritative source)	82.2431	88.3534	87.7356

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

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

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 authoritative 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 without

using an authoritative source. Through this, we show that using an authoritative source 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 for calculating the accuracy, precision and recall. Tables 8 and 9 show the results for the restaurant and cricket datasets, respectively. Figures 1, 2 and 3 show results for the football dataset. Results show that our proposed technique outperforms the current state-of-the-art entity profiling framework. For validating the statistical significance of our experiments, we have conducted a 2-sided paired Student's t -test as described in [49]. For this purpose, we have compared ReLiC with COMET, using the three 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 have been reported in tables 10, 11 and 12. From these tables, it can be observed that the obtained P values are less than α . Further, t values and sample effect sizes are considerable. This shows that ReLiC outperforms COMET with statistical significance.

5.6 Discussion

Various knowledge bases are entirely devoted to the collection of data related to their fields. Owners, organisations or followers regularly update these sources. Thus, the information contained in these sources is generally accurate. To give more importance to the data published by these sources, we have used the concept of authoritative sources for entity profiling. The concept of authoritative

Figure 1. Accuracy values for football dataset.

Figure 2. Precision values for football dataset.

Figure 3. Recall values for football dataset.

source allows such sources to have higher ratings as compared with others. The results demonstrate that the use of authoritative sources is effective for entity profiling. As seen in table 8, the use of an authoritative source in ReLiC has shown an increase of around 11% in accuracy and around 4% in precision and recall.

COMET is a 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 the selection of a 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.

Table 10. Result for 2-sided paired Student's *t*-test using ReLiC and COMET on restaurant dataset.

Metric	<i>t</i> value	<i>P</i> 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

Table 11. Result for 2-sided paired Student's *t*-test using ReLiC and COMET on football dataset.

Metric	<i>t</i> value	<i>P</i> 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 using ReLiC and COMET on cricket dataset.

Metric	<i>t</i> value	<i>P</i> 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

6. Conclusions

In this paper, we studied the performance of various classifiers for entity resolution and found that random forest outperforms the other classifiers. We addressed the problem of entity profiling using a two-phase framework (ReLiC). The first phase (resolution phase) links records to the queries. We associate records to a query using a random forest classifier. The second phase selects the appropriate attribute values from the records to complete a profile. Our results demonstrate that the usage of an authoritative source improves the overall performance of ReLiC. Also, we proposed 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 demonstrate that ReLiC outperforms COMET. ReLiC framework can be further improved by taking temporal information associated with the records into consideration. This can allow the user to get the complete profile of an entity at different points of time. Further, ReLiC and COMET assume the details given in the user query to be highly accurate. However, taking errors in the user query into consideration may further make ReLiC more effective.

References

- [1] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R and Ives Z 2007 Dbpedia: a nucleus for a web of open data. In:

- Proceedings of The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, Busan, Korea, November 11–15, 2007. Berlin–Heidelberg: Springer, pp. 722–735
- [2] Suchanek F M, Kasneci G and Weikum G 2007 Yago: a core of semantic knowledge. In: *Proceedings of the 16th International Conference on World Wide Web*, ACM, pp. 697–706
- [3] Li F, Lee M L and Hsu W 2014 Entity profiling with varying source reliabilities. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 1146–1155
- [4] Li F, Lee M L, Hsu W and Tan W C 2015 Linking temporal records for profiling entities. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM, pp. 593–605
- [5] Wagner A, Barbosa J L V and Barbosa D N F 2014 A model for profile management applied to ubiquitous learning environments. *Expert Systems with Applications* 41(4, part 2): 2023–2034
- [6] Nicoletti M, Schiaffino S and Godoy D 2013 Mining interests for user profiling in electronic conversations. *Expert Systems with Applications* 40(2): 638–645
- [7] Hawalah A and Fasli M 2014 Utilizing contextual ontological user profiles for personalized recommendations. *Expert Systems with Applications* 41(10): 4777–4797
- [8] Guo S, Dong X L, Srivastava D and Zajac R 2010 Record linkage with uniqueness constraints and erroneous values. *Proceedings of the VLDB Endowment* 3(1–2): 417–428
- [9] Calegari S and Pasi G 2013 Personal ontologies: generation of user profiles based on the YAGO ontology. *Information Processing & Management* 49(3): 640–658
- [10] Garcia M M R, Garcia-Nieto J and Aldana-Montes J F 2016 An ontology-based data integration approach for web analytics in e-commerce. *Expert Systems with Applications* 63: 20–34
- [11] Amini B, Ibrahim R, Othman M S and Selamat A 2014 Capturing scholar's knowledge from heterogeneous resources for profiling in recommender systems. *Expert Systems with Applications* 41(17): 7945–7957
- [12] Singla P and Domingos P 2006 Entity resolution with Markov logic. In: *Proceedings of the Sixth International Conference on Data Mining (ICDM)*, pp. 572–582
- [13] Christen P 2008 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*. Wollongong, NSW, Australia: Australian Computer Society, Inc., vol. 80, pp. 17–25
- [14] Fellegi I P and Sunter A B 1969 A theory for record linkage. *Journal of the American Statistical Association* 64(328): 1183–1210
- [15] Bilenko M and Mooney R J 2003 Adaptive duplicate detection using learnable string similarity measures. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 39–48
- [16] Malin B and Sweeney L 2005 *Enres: a semantic framework for entity resolution modelling*. Institute for Software Research International Technical Report, Carnegie Mellon University

- [17] Zhao G, Wu J, Wang D and Li T 2016 Entity disambiguation to wikipedia using collective ranking. *Information Processing and Management* 52(6): 1247–1257
- [18] Fu Z, Zhou J, Peng F and Christen P 2012 A bag reconstruction method for multiple instance classification and group record linkage. In: *Proceedings of the 8th International Conference on Advanced Data Mining and Applications (ADMA)*, Nanjing, China, December 15–18. Springer LNCS, pp. 247–259
- [19] Hu Y, Wang Q, Vatsalan D and Christen P 2017 Improving temporal record linkage using regression classification. In: *Proceedings of 21st Pacific-Asia Conference of Advances on Knowledge Discovery and Data Mining PAKDD*, Jeju, South Korea, May 23–26. Springer LNCS, pp. 561–573
- [20] Christen P 2008 Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*. New York, NY, USA: ACM, pp. 151–159
- [21] Christen P 2007 A two-step classification approach to unsupervised record linkage. In: *Proceedings of the Sixth Australasian Conference on Data Mining and Analytics, AusDM '07*. Darlinghurst, Australia: Australian Computer Society, Inc., vol. 70, pp. 111–119
- [22] Cheng J, Sugiyama K and Kan M Y 2016 Linking organizational social network profiles. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16*. New York, NY, USA: ACM, pp. 901–904
- [23] Derczynski L, Maynard D, Rizzo G, van Erp M, Gorrell G, Troncy R, Petrak J and Bontcheva K 2015 Analysis of named entity recognition and linking for tweets. *Information Processing & Management* 51(2): 32–49
- [24] Christen P 2012 *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Berlin–Heidelberg: Springer
- [25] Talburt J R 2011 *Entity Resolution and Information Quality*. Amsterdam: Elsevier Science
- [26] Gruenheid A, Dong X L and Srivastava D 2014 Incremental record linkage. *Proceedings of the VLDB Endowment* 7(9): 697–708
- [27] Syed H, Talburt J, Liu F, Pullen D and Wu N 2012 Developing and refining matching rules for entity resolution. In: *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, pp. 1–6
- [28] Bhattacharya I and Getoor L 2007 Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data* 1(1): 1–36
- [29] Xiao C, Wang W, Lin X and Yu J X 2008 Efficient similarity joins for near duplicate detection. In: *Proceedings of the 17th International Conference on World Wide Web*, ACM, pp. 131–140
- [30] Benjelloun O, Garcia-Molina H, Menestrina D, Su Q, Whang S E and Widom J 2009 Swoosh: a generic approach to entity resolution. *The VLDB Journal* 18(1): 255–276
- [31] Bilgic M, Licamele L, Getoor L and Shneiderman B 2006 D-dupe: an interactive tool for entity resolution in social networks. In: *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 43–50
- [32] Schewe K D and Wang Q 2014 A theoretical framework for knowledge-based entity resolution. *Theoretical Computer Science* 549: 101–126
- [33] Zhang L, Dong Y and Rettinger A 2015 Towards entity correctness, completeness and emergence for entity recognition. In: *Proceedings of the 24th International Conference on World Wide Web, WWW'15 Companion*. New York, NY, USA: ACM, pp. 143–144
- [34] Prabhakar Benny S, Vasavi S and Anupriya P 2016 Hadoop framework for entity resolution within high velocity streams. *Procedia Computer Science* 85: 550–557 *Procedia Computer Science (International Conference on Computational Modelling and Security, CMS 2016)*.
- [35] Dharavath R and Kumar C 2015 Entity resolution based EM for integrating heterogeneous distributed probabilistic data. *Journal of Systems and Software* 107: 93–109
- [36] Ayat N, Akbarinia R, Afsarmanesh H and Valduriel P 2014 Entity resolution for probabilistic data. *Information Sciences* 277: 492–511
- [37] Hu W and Jia C 2015 A bootstrapping approach to entity linkage on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 34: 1–12
- [38] Yerva S R 2013 *Entities on the web resolution, matching and profiling*. PhD thesis, EPFL, Lausanne
- [39] Wang J, Kraska T, Franklin M J and Feng J 2012 Crowder: crowdsourcing entity resolution. *Proceedings of the VLDB Endowment* 5(11): 1483–1494
- [40] Cheng G, Xu D and Qu Y 2015 C3d+p: a summarization method for interactive entity resolution. *Web Semantics: Science, Services and Agents on the World Wide Web* 35(4 part): 203–213
- [41] Köpcke H, Thor A and Rahm E 2009 Comparative evaluation of entity resolution approaches with fever. *Proceedings of the VLDB Endowment* 2(2): 1574–1577
- [42] Mikolov T, Sutskever I, Chen K, Corrado G S and Dean J 2013 Distributed representations of words and phrases and their compositionality. In: *Proceedings of Advances in Neural Information Processing Systems* 26. Curran Associates, Inc., pp. 3111–3119
- [43] Blanco L, Crescenzi V, Merialdo P and Papotti P 2012 *Web data reconciliation: models and experiences*. Berlin–Heidelberg: Springer, pp. 1–15
- [44] Bradley A P 1997 The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30(7): 1145 – 1159
- [45] Matthews B W 1975 Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) – Protein Structure* 405(2): 442–451
- [46] Hanley J A and McNeil B J 1983 A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 148(3): 839–843
- [47] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay É 2011 Scikit-learn: machine learning in python. *Journal of Machine Learning Research* 12: 2825–2830
- [48] Levenshtein V I 1966 Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8): 707–710
- [49] Sakai T 2014 Statistical reform in information retrieval? *SIGIR Forum* 48(1): 3–12