Journal of Intelligent & Fuzzy Systems 34 (2018) 659–675
DOI:10.3233/JIFS-17946
IOS Press

Entity resolution framework using rough set blocking for heterogeneous web of data

K.A. Vidhya* and T.V. Geetha
Department of Computer Science, Anna University, Chennai, Tamilnadu, India

Abstract. Entity Resolution (ER) is the method of resolving two similar entities used in the process of data cleaning and data integration. However, existing ER Framework lead to exhaustive pairwise comparisons. The most efficient ER method is blocking, inherently uses exponential pair-wise comparisons for the large databases, leading to poor efficiency in resolving the entities. The real world data can either be homogeneous or heterogeneous, generally of two forms, clean-clean ER which does not have any duplicates or dirty-ER which have duplicates within the dataset. Entity Resolution framework is associated with two phases namely the block building phase which construct the blocks where the similar entities are grouped into a single block for effective indexing, while the aim of block processing phase is to reduce the number of redundant pair-wise comparisons. Another perspective is handling of the entity associated with heterogeneous data, in the proposed work the block building phase aims to gather related entities with different representations into a single block with an approximation space. For this purpose semantic-dominance rough set has been used to cluster the attributes of related entities having a varied schema. The similarity between the entities associated with the clustered attributes is determined using a rough-Jaccard similarity measure, grouped to form blocks of varied, but limited size. The pair-wise comparisons between the blocks of entities are carried out only when the lower approximation of the blocks are same, determined by the proposed multi-criteria Pareto optimality, else the entities are not compared, which signifies, the overall number of pair-wise comparisons is reduced. A performance analysis of the proposed technique has been tested on four real-world, highly heterogeneous datasets, and the validation of these algorithms has yielded 99.98% effectiveness and 98.3% efficiency in block comparison when compared to token blocking and attribute clustering methods.

Keywords: Entity resolution, blocking, rough set, heterogeneous data, linked open data

1. Introduction

There has been a constant increase in the quantum of information on the web owing to the growth of online marketing, advertising of commodities, and purchase of online items, all of which inherently create interest in mining useful information from raw data to resolve real-world entities for business development, fraudulent detection of bank transactions, and so on. Linked Open Data is a heterogeneous web of data that collects a massive amount of data connecting common entities, using links in a unified system. The following issues are associated with mining useful information from heterogeneous data a) the data is semi-structured where the schema is highly diverse for the same entities, b) there is a high level of incomplete and inconsistent data, given that it is user-generated, and c) the data is large scale, growing exponentially over time [7]. The chief component of large-scale data integration is entity resolution, which is the task of comparing every profile in the web of data, where comparisons lead to quadratic complexity and do not scale up to the huge volumes of data at hand.

In a heterogeneous web of data, ER is of two types: clean-clean ER, the process of discovering pairs of

*Corresponding author. K.A. Vidhya, Research Scholar, Department of Computer Science, Anna University, Chennai, Tamilnadu, India. E-mail: avidhya06@cs.annauniv.edu.

identical entities in the midst of two large, heterogeneous with overlapping assortments of entities and also duplicate-free [2, 33, 34], and dirty ER, the process that takes inputs as single entity collections with duplicates. Addressing the task of ER for both types requires specialized algorithms as there are specific issues depending on the cleanness of the data. However, dirty ERrequires an additional phase as it has duplicates, an issue yet to be addressed in the literature. Existing blocking methods in the literature are unsuitable for addressing issues in clean-clean ER and dirty ER. A challenging algorithm is required to make ER an easy task in a heterogeneous web of data (Linked Open Data) to address both types of ER in large-scale entity collections. Though there exist several state-of-the-art methods in the literature, the most efficient is blocking, which is an approximation method. The blocking process is either similarity-based or learning-based, where pairwise similarity is computed for consecutive entities, which irrefutably increases time complexity.

The traditional blocking method conflicts with the essential characteristics of a heterogeneous web of data, and adapting most prevailing methods is far from adequate to address our particular concerns: that of achieving maximum effectiveness and efficiency. To accomplish effectiveness, the block building method depends on redundancy, given that the entity is placed in more than one block, significantly reducing the possibility of missed matches. The block size, fixed using the token similarity or attribute clustering method, depends on the key value [35], computed using a hashing method [4] or semantic hash value.

George Papadakis et al. [19] proposed an attribute-blocking method for highly heterogeneous data to address issues in clean-clean ER, but the authors have not discussed the handling of dirty ER. Though the attribute clustering proposed by Papadakis achieves better pair completeness (PC), redundancy ratio value is low, meaning that the number of block comparisons among the blocks is rather high. Blocking techniques are characterized by minimizing the redundancy ratio [42] by means of segregating each description into a single block. Hence, the block building here aims to minimize, positively, the number of comparisons that lead to many missing matches. Therefore, to resolve entities after block building, block post-processing has to be carried out to augment effectiveness.

To address the gaps, a complete framework for entity resolution is proposed using the rough set theory (RST), a statistical tool for addressing uncertainty, attribute selection to aid block building, and block processing. The rest of the paper is organized as follows. Section 2 surveys the related work, Section 3 discusses the preliminaries with the challenges in existing methods, as well as the background in designing an ER framework. Section 4 discusses the proposed work and considers the rough blocking ER and propounds the block processing phase and Section 5 reviews the experimental results and evaluation and Section 6 draws conclusions. Section 7 is devoted to references.

2. Related work

Traditional data warehouses resort to hash-based and sort-based initial blocking procedures. Fellegi and Sunter [1] designed a method to handle tabular data. In their work, they assigned a blocking key for the block in which a description ends up that is also determined by a similarity function on the value of the description for the blocking key. Hernandez et al. [9] worked out a sorted neighborhood method in which first entity descriptions are ordered according to their blocking key values. Gravano et al. [14] proposed multiple blocking key values (BKV) for descriptions by changing every initial BKV into a list of Q-grams, where Q-gram is a substring of Q-characters. P. Christen [2] did an in-depth analysis of the blocking as an indexing technique and listed out the merits and demerits of the blocking methods.

The block key value identification was yet another challenge that needs to be addressed. Shanghoon Lee et al. [32] recommended efficient entity matching using materialized lists where matching is carried collectively using attributes and value definitions. Aizawa and Oyama [7] suggested a suffix-array of BKVs [5], in terms of the sub-strings produced, by eliminating certain of the first characters of the BKV that can be used for blocking removal procedures. Papadakis et al. [21, 22] devised a token blocking method that is a considerably simple approach, relying on the minimal assumption that matched descriptions should, at the very least, have to share a common token.

All the methods above were proposed for a homogeneous web of data and do not address heterogeneous data. In this hash-based method, each distinct token description defines a new block Bt. Papadakis et al. [16, 18, 19] also proposed the post-blocking method called meta-blocking [31] that can reconstruct blocks from a selected block collection for heterogeneous datasets. In contrast to the previous block post-processing procedure, this method drastically discards redundant comparisons, as also comparisons between descriptions that are unlikely to match.

Whang et al. [11] proposed a realistic similarity function method that discovers only a fraction of the matches as well as some non-matches. However, matching the entity descriptions in a highly heterogeneous web of data are usually similar due to their high heterogeneity. Motivated by this study, the Entity Resolution and Blocking Algorithm disregards similarity metrics and distance metrics in favor of the non-metrics used by Araujo et al. [23] and Zhang et al. [13, 24]. Kolb et al. [25, 26] identified a map-reduce using hash-based blocking for implementation and considered the map-phase for descriptions as pairs are being emitted. To minimize the frequency of false matches, Papadakis et al. [17] suggested attribute clustering blocking that can exploit the schematic information of the entity profile from the literature in the web of data.

Further, they projected an alternate view, prefix-infix-suffix blocking that retrieves information with respect to the URI (uniform Resource Identifier) available on the web. Besides, another approach to the string similarity join algorithm initiated by Bayardo et al. [12] builds an inverted index using description values assigned to tokens Rajaraman et al. [15] dealt with reduced numbers of compared descriptions that contain building blocks for sets of tokens appearing collectively in various entity profiles.

Keing and Gal [36] attempted a technique for building blocks based on the maximal frequent item sets proposed by Grahu et al. Isele et al. and Volz et al. [10] proposed the concept of multidimensional overlapping blocks, but the latter [10] worked in the context of the SILK link discovery framework by targeting and implementing links between profiles. McNeil et al. focused their study on considering token blocking and variations in MapReduce [20]. Hassanzadeh et al. [6] adopted and extended existing string matching and semantic matching techniques, and proposed functionality enhancements specifically designed for the link discovery framework. The author showed the effectiveness of the approach in several link discovery scenarios in a real world health care application. Batya Kenig et al. [36] proposed entity resolution based on maximum frequent item sets.

It is understood that the common shortcomings of all the methods above are a result of performance that is dependent on assorted applications in which parameters are to be fine-tuned. All the methods discussed above work for homogeneous data where the schema is well defined. But this assumption does not work well as the web of data is largely heterogeneous, and there is a need for an attribute-agnostic blocking method. At the same time, a good metric space is required for high efficiency and effectiveness of the blocking scheme, not achieved by the existing systems. This work aims for a complete entity resolution framework using a rough set theory, which is a statistical method for resolving uncertainties in block building methods by modeling them as approximation spaces.

3. Preliminaries

3.1. Basic ER model

An entity collection is modeled as a tuple <Ai,Vi,Gi,Ep>, where Ai is the set of attributes and Vi is the set of values and Gi is the set of identifiers and the Ep is the set of entity profiles. An Entity profile is a set of name-value pairs which has an attribute-value association along with it. Let $\mathrm{Ep} = \{\mathrm{e1},\ \mathrm{e2},\ \mathrm{e3}\ldots,\ \mathrm{em}\}$ be a set of entity values $\mathrm{S} : \mathrm{EI} * \mathrm{EJ} \rightarrow \{\mathrm{true},\ \mathrm{false}\}$ be a boolean function. The entity resolution of E can be defined as a set of partitions $\mathrm{P} = \{\mathrm{p1},\ \mathrm{p2},\ \mathrm{p3}, \ldots,\ \mathrm{pm}\}$ of E such that S stands for similarity.

$$\forall\ \mathrm{ei},\ \mathrm{ej}\ \in \mathrm{E} : \mathrm{S}(\mathrm{ei},\ \mathrm{ej}) = \mathrm{true},$$

$$\exists p_k \in P : e_i,\ e_j \in p_k,\ \text{ and}$$

$$\forall\ \ p_k \in P,\ (e_i,\ e_j) \in p_k,\ S(e_i,\ e_j) = true$$

Automatically, matching entity descriptions are placed in the same partition P and all the descriptions of the same partition match. The matching entities (S) should introduce an equivalence relation for rough set model, which can be stated among entity description as follows

$$\mathrm{S}(e_i,\ e_j) = true \text{ (reflexivity)}$$

$$\mathrm{S}(e_i,\ e_j) = \mathrm{S}(e_i,\ e_j) \text{ (symmetry)}$$

$$S(e_i,\ e_j) = true\ \Lambda\ \mathrm{S}(e_j,\ e_i) = \mathrm{true}$$

$$\Rightarrow \mathrm{S}(e_i,\ e_k) = true \text{ (Transitivity)}$$

Thus the entities are modeled as entity descriptions of equivalent components in rough set approach. Here P is the set of block partitions, created using the above mathematical model, where the partitions the blocks are represented as are {B1, B2, ..., Bn} for convenience throughout the paper. If there are two entities $(e_i,\ e_j)$ are similar then we say that the profiles belong to the same block thereby reducing the number of redundant comparisons.

3.2. Rough set approach

Basic concepts of crisp RS can be found in [38–41]. From existing literature RS are based on the information system that is a pair IS = (U, A), where U is a non-empty finite set of objects called the universe or corpus and A is a nonempty finite set of attributes.

With all the subsets of attributes S ⊆ A, there is an equivalence relation IND(S):

$$\mathrm{IND(S)} = \{(\mathrm{x},\ \mathrm{y}) \in \mathrm{U}^2 | \forall \mathrm{a} \in\ \mathrm{S},\ \mathrm{a(x)} = \mathrm{a(y)}\} \quad (1)$$

The partition of U generated by IND(S) is denoted

$$\mathrm{U/IND(S)\ or\ (U/S)}$$

$$\mathrm{U/IND(S)} = \otimes \{\mathrm{U/IND(\{a\})} | \mathrm{a} \in \mathrm{S}\} \quad (2)$$

Where

$$\mathrm{A} \otimes \mathrm{B} = \{\mathrm{X} \cap \mathrm{Y} | \forall \mathrm{X} \in \mathrm{A},\ \forall \mathrm{Y} \in \mathrm{B},\ \mathrm{X} \cap \mathrm{Y} \neq \phi\}$$

If (x, y) ∈ IND(S), x and y are indiscernible by attributes from S. For any selected subset of attributes S, there will be sets of objects that are indiscernible based on those attributes. These indistinguishable sets of objects therefore define an equivalence or indiscernibility relation, referred to as the S-indiscernibility relation. However, the target set X can be approximated using only the information contained within S by constructing the S-lower and S-upper approximations of X:

$$\underline{\mathrm{S}}\mathrm{X} = \{\mathrm{x} \in \mathrm{U} | [\mathrm{x}]_\mathrm{S} \subseteq \mathrm{X}\} \quad (3)$$

$$\bar{\mathrm{S}}\mathrm{X} = \{\mathrm{x} \in \mathrm{U} | [\mathrm{x}]_\mathrm{S} \subseteq \mathrm{X} \neq \phi\} \quad (4)$$

$$\mathrm{POS_S(Q)} = \cup_{\mathrm{X} \in \mathrm{U/Q}}\ \underline{\mathrm{S}}\mathrm{X} \quad (5)$$

$$\mathrm{NEG_S(Q)} = \mathrm{U} - \cup_{\mathrm{X} \in \mathrm{U/Q}}\ \bar{\mathrm{S}}\mathrm{X} \quad (6)$$

$$\mathrm{BND_S(Q)} = \cup_{\mathrm{X} \in \mathrm{U/Q}} \bar{\mathrm{S}}\mathrm{X} - \cup_{\mathrm{X} \in \mathrm{U/Q}}\ \underline{\mathrm{S}}\mathrm{X} \quad (7)$$

In summary, the lower approximation of a target set is a strict approximation consisting of only those objects which can positively be identified as members of the set. The upper approximation is a vague approximation which includes all objects that might be members of target set. The positive, negative and Boundary region for RS is given above.

3.3. Problem statement

Of two individually clean datasets DB1 and DB2, the profiles P1 and P2 are said to match only if they refer to the same real-world entity, in which case the two profiles are said to refer to a similar profile and P1 ≡ P2 denotes the same. The eventual goal of the blocking method is to isolate matching entity profiles effectively with high recall and efficiently with fewer comparisons, as far as possible. Efficiency and effectiveness usually decide the performance of the proposed blocking method. Efficiency can be defined as the number of pairwise comparisons that a set of blocks requires, proportional to the cumulative number of blocks resulting from the same collection, B. The set of detected pairs of matching entities, DPB, influences effectiveness to say that a single comparison is made between the blocks.

Pair completeness (PC) emphasizes a matching pair count of entities, reckoned at least a block in common, or else it is impossible to detect [3]. It is defined as PC = |DPB|/|Be1 ∩ Be2|, where |Be1 ∩ Be2| denotes entities shared by E1 and E2, defined the gold standard. The higher the PC percentage portrays the blocking scheme, the higher it's effectiveness. Comparisons within the block pool, with respect to the gold standard defined, measure the reduction ratio (RR).

The higher the redundancy ratio denoted, the higher the efficiency of the blocking pattern. The work emphasizes an approximation technique for the blocking method which addresses individually clean and dirty ER, discussed as follows. The problem we are trying to address can be defined as finding similar entities in accordance with the baseline datasets to resolute the same real-world entities, given two entity collections with or without duplicates.

The typical trade-off between the two measures is that the less the number of comparisons within the blocks, the more effective the blocking scheme. However, while the number of detected pairs will be more, it diminishes the efficiency of the blocking scheme and vice versa. On the other hand, in the existing literature, attribute clustering followed by supervised meta-blocking leads to a time complexity of O (N1+N2), which can be overcome by the proposed method with minimal times of comparison as well as maximized reduction ratio.

4. Proposed work

To address the gaps, a novel entity resolution framework has been proposed using rough set theory for addressing large databases like clean-clean ER and dirty ER, over a heterogeneous web of data. Pay-as-you-go ER framework [36] exploited a rough estimate for finding the likelihood of matching entities to form initial hints. Using the hints the records are grouped into similar real-world entities. The usage of the hints does depend on the strategy of the ER algorithm, not all algorithms are amenable in using these hints. The construction of hints cause an additional overhead. In our work we have proposed the rough approximation technique, using the classical rough set theory where the approximation is done only for the related entities to the lower approximation, using the indiscernibility measure and proposed rough-Jaccard similarity measure.

The proposed ER framework has two phases like block building and block processing, the block building aims to index the blocks effectively and the block processing technique tackles the issue of redundant comparisons to improve the efficiency. In block building phase for deciding the optimal blocking key criterion, we have proposed the semantic dominant rough set which addresses the issue of heterogeneity. The Local Sensitive Hashing blocking method [37] aims to find the nearest neighbor in a high dimensional space, where the similar items map to the same buckets with high probability. The probability is calculated using the semantic and structural similarity for the short text, by constructing the taxonomy tree. However, the proposed work aims to achieve attribute level similarity that has varied representation of similar attributes, to select an optimal set of blocks to be created. Since the data is heterogeneous the schema of the attribute varies so using the distance measure semantically similar attributes are clustered [29, 30]. This has been used as the pre-processing step to improvise the blocking method to tackle the problem of heterogeneity and identify the similar attributes to be indexed.

Once the number of optimal blocks to be clustered are identified, the rough blocking method will enhance the standard blocking method into an approximate blocking method. The approximation space is designed using the proposed mathematical model. The similar entities are identified using rough-Jaccard similarity where the related entities are placed in an approximation space. For effective block building scheme, rough set blocking algorithm approximates overlapping blocks with the proposed optimal similarity, applied to reduce the number of pair-wise comparisons concisely by keeping the block size as minimum as possible with related entities. The logic behind this methodology is that a large set of exclusively small blocks is extensively more efficient than a set of few, but extremely large blocks that has the same number of block assignments.

In ER framework the major guidelines are to detect the duplicate entries, superfluous records and non-redundant entities to eliminate them. To achieve this, a novel block processing technique using Pareto optimal rough set has been proposed which uses comparison scheduling technique to reduce the redundant comparisons. The number of pair-wise comparisons is much reduced in the proposed method, as the number of comparisons of the entities will only be done within the blocks if the lower approximation is similar. The blocks with different lower approximation will not be compared thereby increasing the efficiency of the blocking method. This processes all the entities residing order in multiple blocks without disseminating the significance of the same object residing in other blocks.

4.1. Existing dominant rough set

The existing work discusses the mathematical background used in the dominant rough selection of attributes to be blocked. The dominance-based rough set theory uses four categories of information in the form of an information table. The dominant rough set models are S = <U,Q,V,f>, where U is a finite set of objects called the universe and represented by the rows of the table; Q = {q1, q2, . . . , qm} is a finite set of characteristics - the columns of the table where V is the domain of characteristics ‘q‘, expressed in the form: $V = U_{q \in Q},\ V_Q$ and f is the information function assigned to each pair: and object x – characteristic q, such that $f : U \times Q \rightarrow V$ with $f(x,\ q) \in V_q,\ \forall q \in Q,\ x \in U$ [8]. The set of characteristics is composed of criteria $C^{>}$, i.e., characteristics with preference-ordered attributes $C^{=}$, i.e., characteristics with non-ordered attributes.

4.1.1. Proposed dominant rough set

Once the attributes are clustered, block building becomes easier. We have proposed a method to find the dominant features to be placed in rough blocks, thereby reducing the number of comparisons required between blocks, with a significantly lesser number of overlapping entities. The preference dominant relation is used to identify the dominating set.

Algorithm Dominant Rough Attribute Selection (EP_A,Dpos,Dneg,Sim)

```
Input: A set of clean-clean EP = {$EP_1$, $EP_2$, ... $EP_n$}
Output: A set of Rough Attribute Clusters
  $EP_A$ = {$EP_{A1}$ ...... $EP_{An}$}
$D_{pos}$: Positive Region DNeg: Negative Region $TN_i$: Token Names
  of Entity 'i' S: Semantic Similarity
  CS $\leftarrow \phi$ // ClusterSet
  $C_{ij} \leftarrow \phi$
  //Finding all the dominant clusters using Rough Set
  for $\forall$ Ep do
     for each $T_i \in R(T_{Ni})$ do
        $C_{ij} \leftarrow$ getSimValue (A1,A2)//word similarity
     end for
        CS $\leftarrow C_{ij}$ //Total number of clusters
  for $\forall$ CS do
     for each $\overrightarrow{x_i}$ and $\overrightarrow{x_j}$ be two clusters
        If $\overrightarrow{x_j} \succeq \overrightarrow{x_i}$ then
           $D_{pos} :\leftarrow \overrightarrow{x_i} \succeq$ .// $\succeq$ — Semantic Dominance Relation
           $D_{neg} :\leftarrow \overrightarrow{x_j}$.
        end for.
     end for.
     $D_{ij} = D_{ij} + 1$ // DominanceClusterCount
  end for //Dominance Clusters
  Index all the cluster CI $\leftarrow D_{ij}$
  Rough Information System (U,<Dpos,DNeg>,IF,S)
```

Consider two heterogeneous datasets, DB1, which is a combination of two similar datasets but from different sources where the schema varies as input. Entity profiles EP_1 and EP_2 have incompatible schemas, and resolving the problem of heterogeneity depends on finding the semantic similarity between attributes $EP1_A \equiv EP2_A$. Semantic dependence is calculated for the entire dataset and similar tokens clustered to achieve it. DC_{ij} is the set of dominant clusters formed for datasets with similar data. The Dpos is a positive region with similar clusters and Dneg the region where attributes are dissimilar. Ep is the entity profile with $EP2_A$ the attribute corresponding to the single-entity profile. Once the attributes are discriminated, blocking becomes an easier task. Let us see the working of the dominant feature selection using the following example from state-of-art literature,

e1 = (Name, Two Tars) (starring, Arthur Stanley Jefferson) (writer, H.M. Walker)(starring, Oliver Hardy) (writer, Thomas Leo McCarey)

e2 = {(title, Kill or Cure), (showcasing, Shashi Kumar, Doddanna, Shruti, Tara, Dwarakish)}

e3 = {(MovieName :: A Brighter Summer Day) (Author, Edward Yang, Larry Fine) (starring, Larry Fine, Arthur Houseman)}

e4 = {(title, Filmfare Best Movie Award), (starring, Stuart Whitman)}

e5 = {(Name, Godzilla), (featuring, Kunio Murai. Fujita J., Shinichi) (director,R.King)}

e6 = {(MovieName, Two Tars), (LeadRole, Stanley Jefferson, Hardy) (director, H.M. Walker Thomas McCarey)}

e7 = {(MovieName, Godzilla vs. Gigan) (Starring, Kunio Murai, Zan Fujita, Shinichi Sekizawa) (director, Robert King, Thomas)}

Semantic Dominant Attribute Clusters
C1 = {(title, MovieName, Name)}
C2 = {(featuring, starring, LeadRole)}
C3 = {(author, writer, director)}.

Using the comparisons above, entities e1 and e6 are found to be similar entity profiles representing the same real-world entities. Dissimilar representations of the same attributes - such as movie names, names, or titles - are combined as a single cluster and, likewise, other clusters constructed. However, since synonymous matching alone could not fetch appropriate cluster entities, the semantic dominance of attributes is considered to facilitate the clustering of similar attributes. From the above example, we can infer that the Cij in the algorithm forms initial clusters randomly as word clusters but since we are addressing the clean-clean ER and dirty ER which apparently holds a different representation of the same word. This leads to the necessity for computing the semantic relatedness, to address this issue we are proposing the dominance relation where two clusters are compared using the structural similarity and linguistic similarity.

Dpos region is constructed with semantic dominance relation which takes C1, C2, and C3 as semantic rich clusters which help to block similar entities. The above example, apparently shows that the attributes are schema-agonistic attributes which are grouped based on the semantic similarity. In literature, the LSH semantic measure has been calculated using the structural and linguistic similarity, but the proposed work considers the semantic measure, as a pre-processing step for effective blocking. The proposed work make sure that all the numeric blocks are indexed individually according to values, whereas the ordinal variable is indexed according to the co-occurrence of the text values are approximated using the similarity measure [28], thereby significantly reducing the pair-wise comparisons. Figure 1 shows the workflow process of the proposed work.

4.2. Rough set block-building ER

The block building phase is used to cluster similar entities into blocks by rendering the task of entity

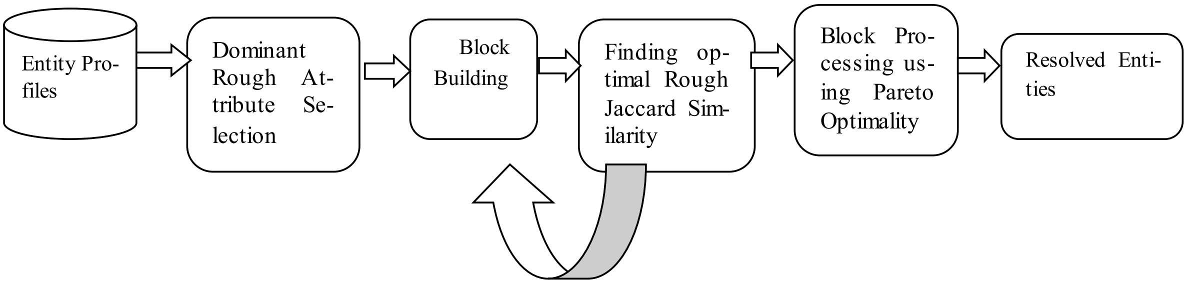
Entity Profiles
Dominant Rough Attribute Selection
Block Building
Finding optimal Rough Jaccard Similarity
Block Processing using Pareto Optimality
Resolved Entities

Fig. 1. Rough set based blocking method.

resolution scalable. Blocks are created according to the blocking strategy that consists of two parts: the lower approximation function and similarity computation (also called the transformation method) (F_t), and the boundary approximation function (or the constraint function) that comprises conditions for placing entities into blocks (F_c). For each block b_i there is a decision attribute modelled in the rough set theory $d_i^c \in D_c$ that decides whether tokens of the entity profile have to be placed in the block. The rough blocking method is proposed with the indiscernibility measure which eases the blocking strategy to fix the number of rough blocks.

This work proposes the optimal Jaccard similarity measure for entities with a threshold value of 0.5 to 0.8, and the two given entities ($\mathrm{E_{vi}}$, $\mathrm{E_{vj}}$) compute the Jaccard similarity ($\mathrm{E_{Vi}}$, $\mathrm{E_{Vj}}$)∀ tokens in $\mathrm{E_{ij}}$. The optimal Jaccard similarity is calculated for blocks with rough entities. Co-occurring entities are assigned to the same block, with the optimal similarity value between the tokens of the entities. The similarity between the two sets is calculated using the Jaccard similarity measure and when there are dissimilar objects, subset similarity is calculated using the proposed rough-Jaccard similarity.

The preference order of these attributes for the blocking scheme is decided using the dominant rough theory, and the blocking modeled as rough blocks.

$|y\backslash T|$ denotes the subsets which have closer proximity to the lower approximation region. Ω denotes the approximation measure in which discernible items are assigned. Once the lower approximation is fixed the tokens that have the same representation, forms the upper approximation entries and thus the blocks are modeled as rough blocks to reduce the number of blocks. Let us try to understand the rough blocking method with the above given example. For example, McCarey is the name that has to be indexed in the lower approximation and Thomas Leo is the surname for McCarey, Thomas Leo is the same name as Thomas. The aim of this work is to enhance the individual block with Mccarey as a lower approximation and Thomas Leo, Thomas as upper approximation element where these will be the subsets of the rough block. Instead of regular jaccard similarity, the porposed rough-Jaccard similarity has been used for forming the minimum number of subsets related to the entity. This is represented as $\underline{A}(y) \subseteq O \subseteq \bar{A}(y)$ where O is the optimal similarity.

Optimal Similarity Block Building Algorithm OSBB (Ei, $A(y)$, $\bar{A}(y)$, Osim, Ω)

```
Input: Set of Entities $E_i$, i = {1 . . . n}
Output: Blocked Entities < $\mathrm{B_{1L}, B_{1U} \ldots\ldots B_{NL}, B_{NU}}$ >
  T = {$\mathrm{t_1 \ldots t_1}$}
    for ∀ $\mathrm{E_i, E_j \in E_{ij}}$
      compute JaccardSimilarity ($\mathrm{E_{ti}, E_{tj}}$)∀ T in $\mathrm{E_{ij}}$.
      $\mathrm{L_B}(\Omega) \leftarrow (\mathrm{E_{Vi}, E_{Vj}})$ if $\mathrm{J_{sim}} > 0.9$
      For $\mathrm{O_{sim}}$ >threshold [0.5 to 0.8]
      L1: for ∀ T ⊆ $U$ and every O ∈ $O_{sim}(y)$ then
  $\underline{A}(y) \subseteq O \subseteq \bar{A}(y)$
      compute If $|y \cap T| \geq |y\backslash T| \Leftrightarrow \frac{|y\backslash T|}{|T|} \geq 0.5$
        $\Rightarrow sim(T,\ O \cup y) \geq Sim(T,\ O)$
        $\mathrm{O} \in A(T) \wedge (\forall y \in B_{ij}(T),\ T \subseteq O)$
          $\frac{|y\cap T|}{|y\backslash T|} \geq \frac{|T\cap O|}{|T\cup O|} = Sim_J(T,\ O) = \Omega$
      $\mathrm{U_B}(\Omega)$ Block ← Eij ($\mathrm{O_{SIM}}$)
      Repeat L1 until all entities are assigned in blocks
      $|y\backslash T|$ is the set of subsets in T.
      Block $\mathrm{B_{ij}} \leftarrow \mathrm{L_B}(\Omega) + \mathrm{U_B}(\Omega)$
      End for
```

This best strategy in this rough blocking is that the pair-wise comparison is done only when the lower approximation are the same for two blocks. This drastically reduces the with-in block comparisons.

Definition 1. Block Collection $\mathrm{B_i}$, can be defined as follows, the definition of the block collection is equal to the sum of the lower and upper approximation blocks, given by

$$BC_{IND}\mathrm{E}_j = \frac{\sum_{e\in E_j} b_{jl} \in B + \sum_{e\in E_j} b_{ju} \in B}{|E_j|} \tag{8}$$

BC is the blocking cardinality which is defined for individual entity $\mathrm{E_j}$, in the entity collection where j can take a value of 0 or 1 entity present or not respectively.

If there resides an entity Ej we have, to sum up, the $b_j i$ (lower approximation block of element j) and $b_j u$ (upper approximation block of element j) which are the cardinalities of both lower and upper blocks respectively. The summation is given as the blocks of the lower and upper approximation for any single entity in a block. Once the individual cardinality has been defined, then overall approximation has to be defined to fix the block size of all the similar entities.

Definition 2. Block Collection $\mathrm{B_i}$, the overall block size, can be defined by the following equation. The $\mathrm{BC_{OV\text{-}app}}$ is the overall blocking which takes into account where $e \in E_j$ the entity, 'b' the block, is and Eij the block index which holds the list of blocks an entity resides in.

$$BC_{OV}E_{ij} = \frac{\sum_{e \in E_j} b_{iu} \in B}{|E_{iu+iL}| + \left|E_{ju+jL}\right|} + \frac{\sum_{e \in E_j} b_{il} \in B}{|E_{iu+iL}| + \left|E_{ju+jL}\right|} \tag{9}$$

The rough blocking method is illustrated using the above block diagram, storing the respective information for the blocking process, carried out for each and every block. The individual block cardinality and the overall block cardinality metrics are checked to determine the proposed rough blocking quality. Consequently, the block building is idealized as given in the above reference. Next phase is block processing phase where the internal entities are resolved within the block. From the literature there are two metrics to be addressed for evaluating the effectiveness and efficiency of the proposed system. Pair completeness and duplicate blocks are considered to check the quality of the blocking scheme strategy.

4.3. Block processing phase

After the entities have been assigned to individual blocks, the significance of an entity residing in each block has to be ascertained to find similar blocks. Nevertheless, Entity Resolution framework is complete without the block processing phase, which handles the redundant blocks, redundant-negative blocks, and superfluous block comparison carried out to resolve the similar entities. To achieve this we have proposed Pareto optimality motivated from the efficient room-sharing problem. Given a list of blocks, it determines the Pareto optimal set, which is the feasible set for two given entities, and the exact matching blocks are returned. Adapted from the literature we have proposed the following block processing technique which achieves good recall measure when compared to token blocking and attribute blocking method.

Given two entity profiles $e_i, e_j \in B^n$, we say that $e_i \equiv e_j$, if the number of blocks of $e_i = e_j$ for $<$ i, j >= 1 ... k. The vector of decision variables $e_i \in B \subset B^n$ is non-dominated with respect to B, if there does not exist another $e_i^{'} \in B$ such that $f(e_i^{'}) \prec f(e_i)$. The vector of decision variables $e_i^{'} \in F \subset B^n(F) \rightarrow$ is the feasible region called the Pareto optimal, if it is not dominated with respect to F.

The dominance criteria are fixed by the lower approximation blocks. Suppose, say, that two entities have exactly matching lower and upper blocks, the feasible region is said to be positive. Suppose the two entities have the same lower boundary values, the feasible region is partially similar. If the lower and upper blocks do not match, the entities are not the same. If there exists a feasible region between two entity profiles, it is said to represent the same real-world entities.

4.3.1. Block processing

Definition 3. (Block Purging Technique)

Similar entities are purged, merged with the respective blocks, and the number of similar entity descriptions grouped together. Thus the resolution of entities is done in O (n) time, where n is the number of lower and upper entities in a block.

Positive Feasible Region → e3, e4 [similar upper and lower approximations]

Partial Feasible Region → Partially similarity [constrained feasibility]

Negative Feasible Region → Not overlapping [different entities]

The entity of decision variables

$$\mathrm{e1} \rightarrow \mathrm{B1_U},\ \mathrm{B3_L},\ \mathrm{B5_U}$$

$$\mathrm{e2} \rightarrow \mathrm{B2_L},\ \mathrm{B3_L},\ \mathrm{B4_U}$$

$$\begin{array}{ll} \mathrm{e3} \rightarrow \mathrm{B2_L},\ \mathrm{B7_U} & \\ & \textit{Positive Feasibility Region} \\ \mathrm{e3} \rightarrow \mathrm{B2_L},\ \mathrm{B7_U} & \end{array}$$

$$\mathrm{e5} \rightarrow \mathrm{B2_U},\ \mathrm{B5_U}$$

The entity index of block propagation.

There are two types of block to be compared and removed, superfluous blocks and redundant blocks. The Block purging is done using the feasibility region, where the positive region decides whether the two entities are the same. If so, they have to be merged into a similar block along the entity identity tag. If the two entities are dissimilar having different lower approximation blocks and similar upper approximation blocks, they are defined as partially similar blocks or share constrained similarity. Depending on the levels of block similarity, the partially similar block is merged when the similarity between the upper approximations is exactly the same as constrained feasibility. Another proposed region is an entirely negative region, where the lower and upper approximations are exactly dissimilar.

Pareto Optimal Rough Block Processing (PORB) (Bn,e_i,PS,F)

```
Input: Set of Blocks {$B_{1L} \ldots B_{NU}$} //redundant and non-match

Output: Set of Unique Resolved Entities
  $\forall$ entities with $B_1$ to $B_n$
     Get the number of Blocks that has an entry for all e1 . . . en
     //For finding the similarity among the blocks which is
  multi- criteria decision-making
  For any two entity description
Find the feasible Pareto optimal set such that $f(e_i^{'}) \prec f(e_i)$.
         The values of decision attribute
         $e^* \in F \subset B^n(F) \rightarrow$ Positive Feasible Region $\rightarrow$ Ps
  (Pareto optimal)
         If there exist a Pareto optimal set <Ps> of two decision
  criteria
               Then $e_i \equiv e_j$ [Similar]
         else
               $e_i \neq e_j$
         end If
            end For
```

The internal block analysis is done by identifying the feasible block for each entity as well as scrutinizing the description of each block within the resource. The Pareto optimal rough set has been proved to be efficient and is discussed with the evaluation results.

5. Experiment evaluation

The aim of the experimental evaluation is to analyze the effectiveness and efficiency of the proposed framework with the existing state-of-the-art methods. Standard metrics, listed in the following section, are being used to analyze the blocking algorithm with varying characteristics of entity descriptions in Linked Open Data datasets. The set of attributes has been selected to test the performance of the algorithms and evaluation is done to penalize the quality of the proposed approach.

First, we compared the proposed work with the token blocking and attribute clustering blocking methods (as baseline methods) to check its effectiveness. Thereafter, to check the efficiency of the ER framework, the block processing strategy is evaluated for all block building methods. Individual comparisons are carried out as against the established ground truth specified in standard datasets.

5.1. Experimental setup and datasets

The work is carried out in an Intel core i5 with a 16 GB RAM since the processing of the DBPedia dataset requires a much higher RAM configuration. Knowledge bases like DBpedia are derived from a specified link reference given below, Wikipedia, and information pertaining to entities is extracted from these. In our experiments, we have used DBPedia and Infoboxes, which initially originated from DBPedia (versions 3.7 and 3.5 respectively). Other datasets used include the BTCDBPedia and BTCRest.

5.1.1. Dataset description

Table 1 gives the statistics of the RDF triples and entity descriptions which gives the description of average attribute-value pairs per description, attributes, entity types and attribute/entity types. The number of duplicates in the dataset. The table above lists the standard datasets available, of which certain are clean-clean and others have duplicate values. The datasets are combined to form the entity collections presented in Table 2. The DB1 combines data from Infoboxes with the BTC12DBpedia[1], a naturally heterogeneous collection. The dataset has a vast collection of n-triples and attributes, with a number of matches in entities with other neighboring entity profiles. The number of n-triples is much higher in the dataset thus forming and profiling it as entity profiles is an arduous task, while the proposed ER framework tackles this entity comparison in a profound way.

DB2 combines BTC12DBPedia with BTC12Rest[2] and, created from several dissimilar datasets, results in a heterogeneous collection.

The average number of attributes in an entity description is much higher. DB3 combines DBPedia movies with IMDB[3], as originally used in [16]. It is a heterogeneous collection with both datasets containing descriptions only of movies. DB4 combines LinkedCT and Diseasome, where RDF triples of biomedical data are combined with the respective diseases to form a disease profile in Diseasome. Entity profile matching is done and it is clearly a homogenous set in the collection as the two sets only have the disease as an attribute in common. Diseasome[4] has a total of 372 links to LinkedCT[5]. The following table gives the statistics of a number of attribute-value pairs and comparisons without blocking, using the clean-clean ER and dirty ER datasets.

5.1.2. *Evaluation measures*

We have used three measures to evaluate the proposed work on blocking methods: performance, techniques, and metrics. The traditional pair of completeness and redundancy ratio are being used to compute effectiveness and efficiency and address both in the proposed system. Technical metrics address internal block processing measures in terms of how blocks are arranged, what the average number of comparisons involved are, and the number of blocks that have been reduced due to the rough optimal similarity measure.

5.1.3. *Experiment results*

Experiments are carried out for the proposed system with the available standard datasets and testing done with the gold standard. The datasets used for implementing and testing the system, downloadable from the links above, are displayed in Tables 1 and 2.

5.1.4. *Performance of dominant attribute clustering*

To address the gap of heterogeneity in the web of data, the system forms dominant clusters as discussed in section 3, for evaluation of the results the system has been tested with the baseline method, entity resolution using attribute clustering, and the proposed dominant clustering method where semantic dominance is addressed. Figures 2 and 3 make it evident that precision is high but recall value is low in attribute clustering, demonstrating that the proposed method works better with the dominance is addressed. The following figures make it evident that precision is high but recall value low in attribute clustering, demonstrating that the proposed method works better with the baseline approach. It follows, then, that the initial set of attributes with different representations can be uniformly clustered into a single representation, and the blocking done according to descriptions of similar entities.

5.1.5. *Evaluation of blocking characteristics*

To address the gap of proposed block building method we have compared the blocking characteristic of the token blocking, attribute clustering (AC) and prefix-infix method [2] with the rough blocking method. The rough set blocking method is verified with standard metrics as well as the ones proposed

Table 1
The linked open data (web of data) datasets

	BTC12 DBPedia	Infoboxes	BTCRest	DBPediaMov	IMDB	LinkedCT	Diseasome
RDFTriples	102306242	27011880	849656	180680	816012	1042536	91128
No of entity	8945920	1638149	31668	1849180	25359	894252	949
avg.attribute value pairs per description	11.44	16.49	26.83	6.54	35.2	12.45	39.45
Attributes	36354	31857	518	5	7	26	7
entity types	258202	5535	33	1	1	92078	878
attributes/entity types	0.14	5.76	15.7	5	7	5	9
duplicates	0	0	863	0	0	0	0

Table 2
Standard Datasets and their statistics

	DB1	DB2	DB3	DB4
RDF Triples	129318122	103155898	996692	1133664
Entity descriptions	10584069	8977588	50797	895201
Avg attribute-value pairs	12.22	11.49	19.62	17.65
Attributes	68211	38872	12	21
Entity types	263737	258232	1	78932
Matches	1564311	30864	22405	7310
Matches including duplicates	1564311	31727	22405	7310
Comparisons (without Blocking)				
clean- clean	$1.37*10^{13}$	$2.83*10^{11}$	$6.4*10^{8}$	$7.89*10^{12}$
Dirty	$5.6*10^{13}$	$4.30*10^{13}$	$1.29*10^{9}$	$1.89*10^{6}$

1,2,3 ← http://csd.uoc.gr/∼vefthym/minoanER/da-tasets.html. 4 ← https://datahub.io/dataset/fu-berlin-diseasome. 5 ← https://datahub.io/dataset/linkedct

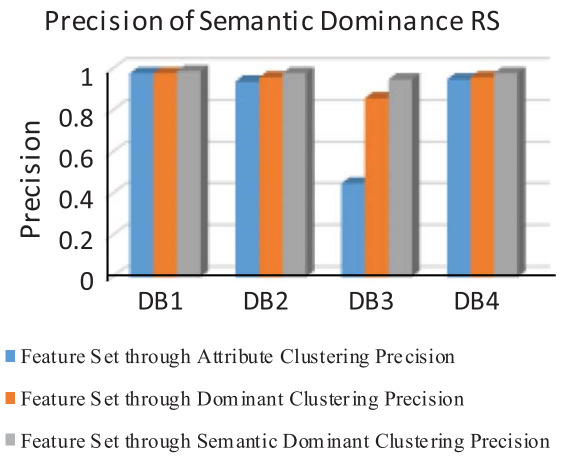

Fig. 2. Performance of Dominant Attribute clustering.

for the optimal block similarity measure. The metrics for the DInfoboxes dataset are given in the following table, which discusses overlapping blocks and block cardinality with baseline methods.

The precision value is high in the semantic dominance, however, this is found to be low in the DB3 why because, the dirty ER has some duplicates within the dataset so the number of clusters to be formed is not much improved, however, the attribute clustering method does not address dirty ER. The recall values of the baseline method are improved drastically by the proposed method as the number of clusters found will be more while using the semantic dominance relation.

The below two graphs discusses the precision and recall measure for the first contribution that procures the semantic clusters from the given heterogeneous attributes. The comparison is made with the attribute clustering method as it works for the heterogeneous web of data. The semantic dominant rough set blocks yield better precision and recall compared to the baseline methods. The block building and block processing characteristics are accounted in the table. From the Table 3, we infer that average number of comparison in the rough blocking method is reduced concisely when compared to other state-of art methods. The three baseline methods, token blocking, N-grams attribute clustering (AC) and attribute clustering has been implemented to evaluate the proposed system.

Tables 3–6 shows the block building and block processing metrics which shows the attribute clusters and the number of blocks needed for each method. The pair completeness and the RR decide the efficiency of the proposed method. Apart from the above metrics, the standard metrics defined rough blocking values like an individual number of blocks and the comparison cardinality is discussed for the baseline methods to analyze the working of the rough blocking method. The rough individual blocking cardinality is the average number of inner blocks that can be placed in the Block B, the individual block has Li(Ω) and Ui(Ω) the approximation spaces. Comparison cardinality is the ratio of the sum of the block sizes and the aggregate cardinality B.

As the number of values semantically related to the lower bound falls into the same block the related

Table 3
Evaluation metrics for DB1

	Block Building Metrics				Block Processing Metrics			
	Attribute Clusters	Number of Blocks	Comparison Cardinality	AVG. Comp	Disk Space	Comparisons	Reduction Ratio	Pair Completeness
Token Blocking	1	1639962	$1.65*10^{-4}$	$5.05*10^{6}$	3 GB	$7.21*10^{12}$	–	98.3%
Q-grams AC	3	5307634	$1.83*10^{-4}$	$0.23*10^{6}$	5.2 GB	$1.78*10^{12}$	68.01%	98.1%
Prefix-Infix	–	3266798	$3.43*10^{-4}$	$7.04*10^{6}$	6.0 GB	$5.39*10^{12}$	92.3%	94.3%
Attribute Clustering	16886	5602644	$1.41*10^{-4}$	$0.24*10^{6}$	5.2 GB	$3.22*10^{11}$	81.09%	98.2%
Rough Blocking	14841	638692	$5.54*10^{-3}$	**$0.21*10^{4}$**	5.8 GB	$1.62*10^{10}$	98%	99.82%

Table 4
Evaluation metrics for DB2

	Block Building Metrics				Block Processing Metrics			
	Attribute Clusters	Number of Blocks	Comparison Cardinality	AVG. Comp	Disk Space	Comparisons	Reduction Ratio	Pair Completeness
Token Blocking	1	122340	$1.98*10^{-3}$	$4.65*10^{5}$	1 GB	$6.18*10^{10}$	–	99.60%
Q-grams AC	3	143263	$2.13*10^{-3}$	$3.23*10^{5}$	1.5 GB	$0.98*10^{10}$	4.43%	99.30%
Prefix-Infix	–	141517	$2.89*10^{-3}$	$7.18*10^{5}$	2.5 GB	$3.45*10^{10}$	92.35%	93.23%
Attribute Clustering	124	150293	$1.78*10^{-3}$	$3.12*10^{5}$	1.5 GB	$4.20*10^{9}$	60.3%	97.21%
Rough Blocking	188	73240	$1.46*10^{-3}$	$2.19*10^{5}$	2.2 GB	$2.62*10^{9}$	94.3%	99.98%

Table 5
Evaluation metrics for DB3

	Block Building Metrics				Block Processing Metrics			
	Attribute Clusters	Number of Blocks	Comparison Cardinality	AVG. Comp	Disk Space	Comparisons	Reduction Ratio	Pair Completeness
Token Blocking	1	40304	$0.65*10^{-2}$	$8.12*10^{3}$	28 MB	$4.18*10^{8}$	–	97.60%
Q-grams AC	3	41930	$0.78*10^{-3}$	$7.01*10^{3}$	52 MB	$2.98*10^{8}$	4.01%	96.30%
Prefix-Infix	–	–	–	–	–	–	–	–
Attribute Clustering	4	43716	$0.59*10^{-3}$	$6.54*10^{3}$	58 MB	$2.21*10^{8}$	30.46%	98.21%
Rough Blocking	4	22340	$0.23*10^{-3}$	$5.54*10^{3}$	64 MB	$0.62*10^{8}$	96.4%	96.32%

Table 6
Evaluation metrics for DB4

	Block Building Metrics				Block Processing Metrics			
	Attribute Clusters	Number of Blocks	Comparison Cardinality	AVG. Comp	Disk Space	Comparisons	Reduction Ratio	Pair Completeness
TokenBlocking	1	22403	$0.90*10^{-1}$	$7.68*10^{2}$	28 MB	$5.83*10^{6}$	–	99.60%
Q-gramsAC	3	27821	$2.23*10^{-2}$	$7.23*10^{2}$	52 MB	$3.28*10^{6}$	5.65%	99.30%
Prefix-Infix	–	19121	$1.12*10^{-1}$	$8.73*10^{2}$	64 MB	$6.29*10^{6}$	–	93.23%
Attribute Clsutering	8	28835	$1.89*10^{-2}$	$6.16*10^{2}$	52 MB	$3.12*10^{6}$	36.92%	97.21%
Rough Blocking	8	17383	$1.28*10^{-2}$	$\mathbf{4.12*10^{2}}$	52 MB	$2.82*10^{6}$	97.4%	99.98%

block count in a single block will be reduced to index and thus the number of comparisons will automatically be reduced. However, the initial design of the block approximation space is quite a challenging task. The following four tables discuss the overall block building metrics and the block processing metrics, discussed in the algorithms mentioned above. Note that the core of rough blocking lies on the initial dominant attribute clustering based on schematic similarity. Attribute definition may vary since it is heterogeneous data, but a word-level schematic dominance is used to cluster attributes of similar representations.

Once this is done, we do a value-based overlap approximation, blocking all similar tokens and keeping the lower approximation token a unique entry across the blocks. Efficient indexing of entities is done side-by-side using the block-id with the lower and upper approximation sets. The lower approximation sets are decided by the equivalence, reflexivity, and symmetry of tokens in matching entities.

The evaluation measure can be divided into two types of metrics the block building and block processing metrics. The dataset DBpedia and Infoboxes entity resolution metrics have been given in Table 3, the value of pair completeness and a number of attribute clusters has been compared with four baseline methods (Token blocking, prefix-infix blocking-grams blocking and attribute clustering). The number of blocks to be indexed is very large, in heterogeneous representation the same attribute can have a different representation, however, the token blocking method considers every attribute as tokens, but in rough blocking method, the initial clusters are illustrated in the table based on the initial clusters the blocking scheme is decided. The blocks are approximated, based the semantically similar entities and the set of blocks are formed to reduce the pair-wise comparisons. The token blocking will have the number of comparisons lesser compared to the prefix-infix method as this requires the URI resolution.

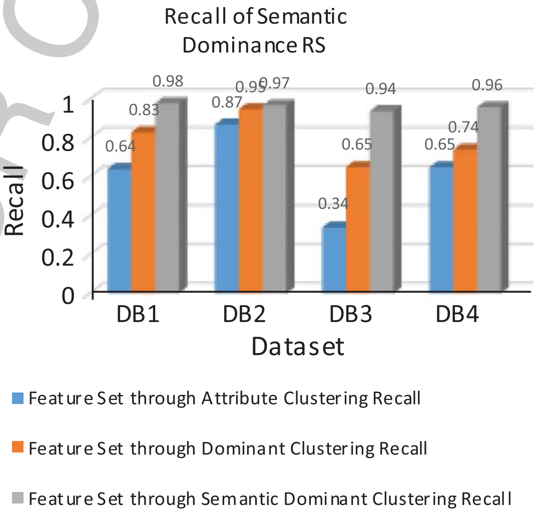

Fig. 3. Performance of Dominant Attribute clustering.

The block-processing metrics like the disk-space, pair completeness and the reduction ratio which takes the token blocking method as the baseline method. The DB2 entity resolution metrics depicted in Table 5, infer that the number of blocks formed in this method is much lesser when compared to the DB1, as the number of similar entities is much more in DB2, another reason is that the number of entity profiles is lesser. The average comparison rate is reduced in the rough blocking techniques in the approximation method the number of blocks in the similar set is much lesser as we are expanding the approximation space for an individual block.

Consequently, the reduction ratio improves from 34% to 85%, along with the pair completeness listed in the table. The metrics for the DB3 IMDB dataset are given in Table 6 which discusses rough blocking method and block cardinality with baseline methods. The BTCRest dataset has 849656 triples with 863 duplicates, comprising a dirty ER dataset with the ground truth specified. The duplicates are initially processed using a two-stage rough blocking method, where the rough blocking is done within the dataset, similar entities are blocked. Once resolved, the resultant entities can be compared with the DBPedia. The building block metrics are compared with every profile in the baseline methods and average comparisons made using the metrics below. However, the dirty ER has 2 times O (nlogn) complexity, as rough blocking has to be initially carried out for blocking similar entities, following which it is compared with the other DBPedia dataset.

We have discussed the metrics for two datasets, and the following table focuses on pairwise comparisons for the clean-clean and dirty ER datasets. The reduction ratio is much improved when compared to state-of-the-art methods, while the number of comparisons within the block is also much higher. The average number of comparisons is reduced in the rough blocking method, as well as the number of duplicates in the block. However, the criticality of the system needs to be adequately adaptable to store the approximation blocking method. The following table listed below gives the number of blocks, number of comparisons, number of reduction ratios for both clean-clean and dirty ER type of datasets. The BTC12Rest dataset has duplicates which are to match with the existing profiles and similar profiles combined. Time complexity in this regard is basically much higher when compared to the clean-clean datasets.

Table 8 shows the other comparison data for clean-clean ER and dirty ER, from the table we infer the token blocking method has a number of comparisons than that of rough blocking method and hence the reduction ratio is considerably improved. However, the attribute clustering does not address the dirty ER. The definition for precision, recall, Redundancy ratio, given in Table 3 are the standard metrics. Recall shows the number of matching descriptions, the blocking method manages to put into at least one common block. The effectiveness of the blocking scheme depends on the recall value. This can be termed as the cost of the blocking scheme. Reduction ratio RR can be defined as the percentage of comparisons that are reduced when the blocking method is applied.

The efficient blocking method should have a lower impact on recall and the same time a great impact on the number of required comparisons. The trade-off is measured by the F-measure namely the harmonic mean of recall and precision. The values of F-Measure are dominated by the value of precision which is much lower in the orders of magnitude lower than those of recall so taking into F-measure alone doesn't suffice the trade-off measure. From the literature, a metric called H3R is used to which is harmonic mean of recall and reduction ratio. Comparatively, H3R measure gives the high values only when the recall and the reduction ratio values are high. In a homogenous entity collection it is not difficult to achieve the recall rate so the more importance is given

Table 7
Evaluation measures

Precision	Precision is the measure of the ratio of the candidate entities to the matched entities. This measures the block quality.	$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$
Recall	A measure of true matches exactly matching the candidate entities mentioned in ground truth.	$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
F-Measure	The Mean of Precision and Recall measure.	$\mathrm{F} = \frac{2(\mathrm{P}*\mathrm{R})}{\mathrm{P+R}}$
RR-Redundancy Ratio	The redundancy ratio is the ratio of number of reduced comparisons (cmp) when blocking is applied.	$1 - \frac{\text{Cmp with blocking}}{Cmp\ with\ BB}$
		BB-Baseline Blocking –Token Blocking
H3R	The mean of recall and reduction ratio.	$\frac{2*(\mathrm{RR}*\mathrm{Recall})}{(\mathrm{RR+Recall})}$
Pair Completeness	Matching entities that occurs in common in more than one block.	$\frac{\lvert DPB\rvert}{\lvert Be1 \cap Be2\rvert}$

Table 8
Comparison of Reduction ratio with clean-clean ER and dirty ER Dataset

Token Blocking	DB1	DB2	DB3	DB4
Blocks	1639962	122340	46304	22405
Comparisons-CleanER	$1.68*10^{12}$	$3.74*10^{10}$	$2.91*10^{8}$	$2.25*10^{6}$
RR-CleanER	88.51%	86.81%	54.50%	84.31%
Comparisons-DirtyER	$5.56*10^{12}$	$3.68*10^{12}$	$2.05*10^{9}$	$2.04*10^{7}$
RR-DirtyER	90.08%	90.87%	–58.85%	90.92%
Attribute Cluster Blocking	DB1	DB2	DB3	DB4
Blocks	5602644	150293	43716	28835
Comparisons-CleanER	$3.22*10^{11}$	$4.2*10^{9}$	$2.13*10^{8}$	$3.28*10^{7}$
RR-CleanER	97.80%	98.25%	6.80%	96.43%
Rough Blocking	DB1	DB2	DB3	DB4
Blocks	638692	73240	22340	17383
Comparisons-CleanER	$0.62*10^{10}$	$2.68*10^{8}$	$1.89*10^{6}$	$3.25*10^{5}$
RR-CleanER	98%	97.54%	76.89%	97.45%
Comparisons-DirtyER	$5.68*10^{10}$	$4.64*10^{8}$	$2.05*10^{6}$	$4.86*10^{6}$
RR-DirtyER	92.08%	94.68%	–44.64%	96.92%

to the reduction ratio but in the heterogeneous entity collection getting more efficient blocking scheme means there should be a high recall so recall rate and reduction ratio, are given equal importance to decide the metrics of the proposed system.

There is a huge difference in DB3 dataset, which contains many more common tokens per description to those of the other collections. Thereby the recall rate is highest for the dataset and also has the maximum matching entities. The Table 8 compares token blocking, attribute clustering and the rough blocking method for clean-clean ER and dirty ER. The average comparisons for the dirty ER and clean-clean datasets have been tested with the baseline method, token blocking. However, the attribute blocking method fails to address the dirty ER dataset, while this method aims to address both datasets with an optimal number of comparisons.

The time complexity of the existing systems and the proposed system are discussed as follows, taking the lead from Jeffery Fisher et al. [27] who proposed split-and-merge block sizes for ER. The similarity-based approach has O (|R|3log(|R|)), while the size-based approach is O(|R|3) [27]. The sorted neighborhood is one of the most efficient blocking algorithms in the literature, yielding an O(nlogn) time complexity which does not consider pair-wise comparisons. For example, StarWorld is a single entity placed in two different blocks, like a star as a separate entity and world as another. The proposed rough blocking reduces the number of comparisons in all the datasets, but for DB3 it has the lesser RR value because of more number of more number of duplicate entities within the data. The time to design the block index and the blocking approximation is quite more than the other method, however, the effectiveness and efficiency of the system are considerably increased.

The SPAN blocking method requires a time complexity of O (Jnlogn), where J is the average number of q-blocks deciding the time complexity of the blocking algorithm. The block indexing and canopy clustering method takes into account pairwise comparisons, which naturally lead to O(n2) comparisons, thus requiring an efficient blocking method to reduce the number of comparisons. In heterogeneous attribute-based clustering, both time and space complexity have an order of O(|N1|.|N2|), where N1 and N2 correspond to the distinct number of unique attributes in E1 and E2 respectively. In the rough blocking method, the time complexity of two given entities is O [|BL+BU|], where BL is the number of blocks in the lower approximation and B the number of blocks in the upper approximation. The similarity computation of each token is carried out using O|B| in the overlapping phase. The graph 4, 5, 6, 7 discusses the precision, recall, f-measure and H3R (Harmonic mean of recall and reduction ratio) values as per the definition are given in Table 7. The precision graph for the rough blocking method is shown in Fig. 4.

The graph shows that the evaluated methods manage to greatly reduce the number of comparisons that would require if the blocking is not applied. The graph above discusses the F-Measure, recall and harmonic mean values across four datasets. Figure 4 discusses the precision graph for the baseline and proposed a method which yields $1.56*10^{-6}$ for token blocking method and $0.93*10^{-5}$ for rough blocking method in clean-clean DB1, $1.97*10^{-3}$ and $7.64*10^{-4}$ for the both methods in DB4 dataset respectively. Figure 5 discusses the recall graph

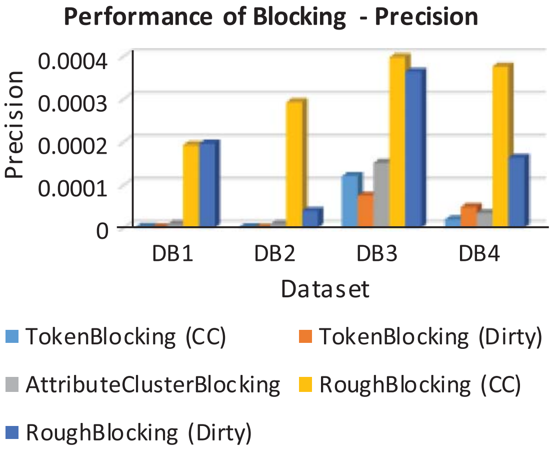

Fig. 4. Performance for Blocking Method Precision Values.

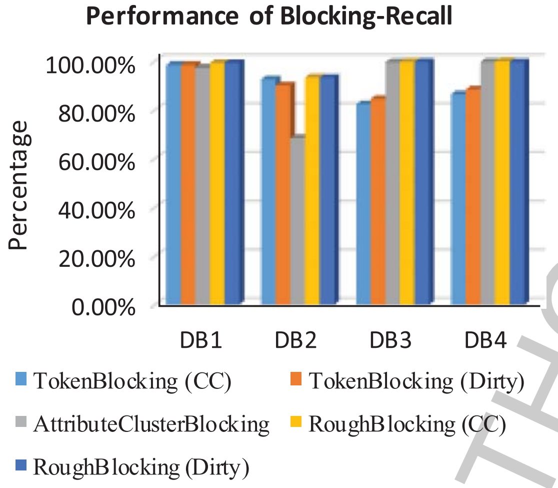

Fig. 5. Performance for Blocking Method- Recall Values.

which implies that the rough blocking method consistently has high recall values when compared to the other baseline approaches. There are more duplicate entries in DB3, leading to a drop in the redundancy ratio value w. Inferences from the graph show that the rough blocking method has gained a high harmonic mean as the redundancy ratio of the rough blocking method is high, compared to the other values. The precision value of the rough blocking method is better, compared to state-of-the-art methods like token blocking and attribute clustering.

The DB3 initially contains a much number of smaller number of comparisons and a higher ratio of matches to non-matches, so the reduction ratio of this case is limited and hence the H3R is N/A for this cases. The recall achieved by the token blocking

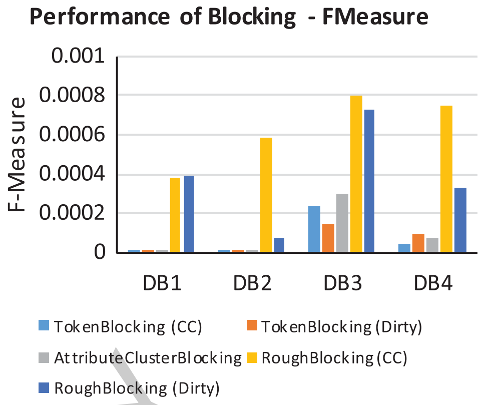

Fig. 6. Performance for Blocking Method –F-Measure Values.

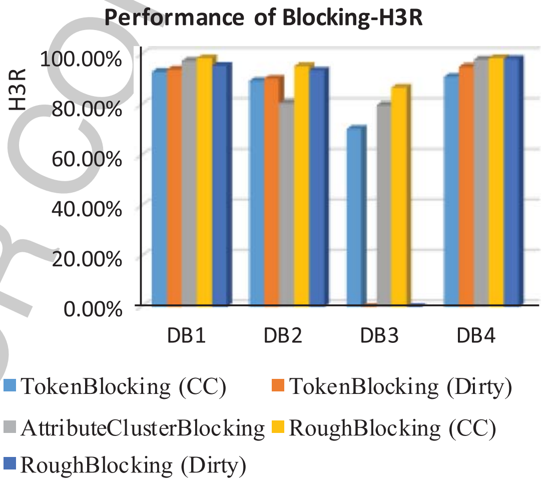

Fig. 7. Performance for Blocking Method –H3R Val.

method is very high when there are more overlapping entity profiles and the gain in precision brought by token blocking method are high while incurring a low cost in the recall. The prefix-infix blocking method can improve both the precision and recall for the clean-clean ER but the values are declined when the dataset is dirty ER.

The Fig. 4 graph discusses the precision of all blocking method where the datasets shows the consistent increase in precision value emphasizes the effectiveness of the proposed approach has been improved which avoids the superfluous comparisons.

From Fig. 5, it can be inferred that the recall of all state-of-art methods are compared with the proposed work, which implies the recall value emphasizes the efficiency of the proposed approach meets the existing approaches which avoids the superfluous comparisons.

To conclude, in a LOD cloud, the central entity collection is generally derived from common source which has many similar naming policies and common tokens in that case recall of token and rough blocking method yields high recall whereas the DB4 dataset features a high number of matching to non-matching descriptions, the precision of both the method ranges from $3.64*10^{-7}$ to $2.49*10^{-6}$. The H3R reveals that many of the comparisons that are discarded by blocking in the DB4 dataset correspond to the matches of the entities in the collection.

6. Conclusion

An ER framework has been designed and implemented using a rough approximation method, which works as a similarity score for building blocks in the blocking phase. The method works for clean-clean and dirty ER, using a dominant rough set to find the best matching attributes to be blocked. The dominant cluster results seem to be promising, and the time and space complexities required to implement the blocking method lesser than in existing algorithms. The proposed rough set blocking method has proved to be efficient with 99.8% pair completeness for clean-clean ER and 96.32% for dirty ER, though the time taken to resolve entities will be much higher going into a two-phase rough block method. As far as the block processing phase is concerned, the removal of redundant blocks has been done using the Pareto optimal rough set where there will be a multi-decision analysis on whether the two entities are indeed similar. The proposed three feasible regions decide whether the optimality falls in the positive feasible region, wherein entities are apparently similar. Evaluation is done using the standard ground truth justified by the experts.

The efficiency of the method has been increased even with approximate blocking method as the reduction ratio is improved compared to the attribute clustering stating that it works better for the large datasets. Even though the blocks are approximated the restriction of the block size is indirectly controlled by the number of the related entities to be inside a block which improvises the reduction ratio of the proposed work. The reduction ratio and the number of iterations required for block purging are reduced by as much as 45%, compared to the baseline methods. As an enhancement of this work, improving the system's time complexity and designing an entity resolution using the map-reduce framework to resolve bio-medical entities have been planned. Another area we would like to investigate is the use of unsupervised learning to match entities and construct an unsupervised block building strategy.

References

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

[2] P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, *IEEE Trans Knowl Data Eng* **24**(9) (2012), 1537–1555.

[3] M. Bilenko, B. Kamath and R.J. Mooney, Adaptive blocking: Learning to scale up record linkage, *In ICDM* (2006), 87–96.

[4] B. Bahmani, A. Geol and R. Sindhe, Efficient distributed locality sensitive hashing, *In CIKM* (2012), 2174–2178.

[5] T. De Vries, et al., Robust record linkage blocking using suffix arrays, *Proceedings of the 18th ACM Conference on Information and Knowledge Management ACM* (2009).

[6] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R.J. Miller and M. Wang, A Framework For Semantic Link Discovery Over Relational Data, *In Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, pp. 1027–1036. ACM.

[7] A.N. Aizawa and K. Oyama, A fast linkage detection scheme for multi-source information integration, *In WIRI* (2005), pp. 30–39.

[8] A. Saxena, L.K. Gavel and M.M. Shrivas, *Rough sets for feature selection and classification: An overview with applications*, 2014.

[9] M.A. Hernández and S.J. Stolfo, Real-world data is Dirty: Data cleansing and the merge/purge problem, *Data Mining and Knowledge Discovery* **2**(1) (1998), 9–37.

[10] J. Volz, et al., Discovering and maintaining links on the web of data. International Semantic Web Conference, Springer Berlin Heidelberg, 2009.

[11] S.E. Whang, et al., Entity resolution with iterative blocking, *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, ACM, 2009.

[12] R.J. Bayardo, Y. Ma and R. Srikant, Scaling up all pairs similarity search, *Proceedings of the 16th International Conference on World Wide Web ACM*, 2007.

[13] D. Zhang, et al., A similarity-oriented RDF graph matching algorithm for ranking linked data, *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on, IEEE*, 2012.

[14] L. Gravano, P.G. Ipeirotis, H.V. Jagadish, N. Koudas, S. Muthukrishnan and D. Srivastava, Approximate string joins in a database (almost) for free, *In VLDB* **1** (2001), 491–500.

[15] J. Leskovec, A. Rajaraman and J.D. Ullman, Mining of massive datasets, Cambridge University Press, 2014.

[16] G. Papadakis, et al., Meta-blocking: Taking entity resolution to the next level, *IEEE Transactions on Knowledge and Data Engineering* **26**(8) (2014), 1946–1960.

[17] P. George, et al., Schema-agnostic vs schema-based configurations for blocking methods on homogeneous data, *Proceedings of the VLDB Endowment* **9**(4) (2015), 312–323.

[18] G. Papadakis, et al., Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking, *EDBT* (2016).

[19] G. Papadakis, G. Papastefanatos and T. Palpanas, Boosting the efficiency of large-scale entity resolution with enhanced meta-blocking, *Institute for the Management of Information Systems* (2015).

[20] N. McNeil, H. Kardes and A. Borthwick, Dynamic record blocking: Efficient linking of massive databases in MapReduce, *In Proceedings of the 10th International Workshop on Quality in Databases (QDB)*, 2012.

[21] G. Papadakis, et al., Efficient entity resolution for large heterogeneous information spaces, *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining ACM*, 2011.

[22] G. Papadakis, et al., Comparative analysis of approximate blocking techniques for entity resolution, *Proceedings of the VLDB Endowment* **9**(9) (2016), 684–695.

[23] S. Araujo, et al., SERIMI: Class-based Disambiguation for Effective Instance Matching over Heterogeneous Web Data, *WebDB* (2012).

[24] D. Zhang, et al., A similarity-oriented RDF graph matching algorithm for ranking linked data, *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on IEEE*, 2012.

[25] L. Kolb, A. Thor and E. Rahm, Multi-pass sorted neighborhood blocking with MapReduce, *Computer Science-Research and Development* **27**(1) (2012), 45–63.

[26] L. Kolb, A. Thor and E. Rahm, Dedoop: Efficient deduplication with Hadoop, *Proceedings of the VLDB Endowment* **5**(12) (2012), 1878–1881.

[27] J. Fisher, P. Christen, Q. Wang and E. Rahm, A Clustering-Based Framework to Control Block Sizes for Entity Resolution, KDD'15, Sydney, NSW, Australia, 2015.

[28] K.A. Vidhya, T.V. Geetha and G. Aghila, Text document classification using rough set theory and multi-level naïve bayes, *International Journal of Applied Engineering Research (IJAER)* **10**(75) (2015), 331–336.

[29] K.A. Vidhya and G. Aghila, Text mining process, techniques and tools: An overview, *International Journal of Information Technology and Knowledge Management* **2**(2) (2010), 613–622.

[30] K.A. Vidhya and G. Aghila, Hybrid text mining model for document classification, *In Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, Vol. 1, 2010, pp. 210–214. IEEE

[31] G. Papadakis, G. Papastefanatos, T. Palpanas and M. Koubarakis, Scaling Entity Resolution to Large, Heterogeneous Data with Enhanced Meta-blocking, *International Conference on Extending Database Technology (EDBT)*, Bordeaux, France: ISBN 978-3-89318-070-7, 2016. on OpenProceedings.org.

[32] S. Lee, J. Lee and S.-W. Hwang, Efficient entity matching using materialized lists, *Information Sciences* **261** (2014), 170–184, Information Sciences.

[33] A. Elmagarmid, P. Ipeirotis and V. Verykios, Duplicate record detection: A survey, *TKDE* **19**(1) (2007), 1–16.

[34] H. Kim and D. Lee, HARRA: Fast iterative hashed record linkage for large-scale data collections, *In EDBT* (2010), 525–536.

[35] B. Kenig and A. Gal, Efficient Entity Resolution with MFI-Blocks, VLDB '09, Lyon, France, Copyright 2009 VLDB Endowment, 2009.

[36] S.E. Whang, D. Marmaros and H. Garcia-Molina, Pay-as-you-go entity resolution, *IEEE Transactions on Knowledge and Data Engineering* **25**(5) (2013), 1111–1124.

[37] Q. Wang, M. Cui and H. Liang, Semantic-aware blocking for entity resolution, *IEEE Transactions on Knowledge and Data Engineering* **28**(1) (2016), 166–180.

[38] K.A. Vidhya and T.V. Geetha, Rough set theory for document clustering: A review, *Journal of Intelligent & Fuzzy Systems* **32**(3) (2017), 2165–2185.

[39] Z. Pawlak, Rough set theory and its applications to data analysis, *Cybernetics & Systems* **29**(7) (1998), 661–688.

[40] N. Yadav and N. Chatterjee, A novel approach for feature selection using Rough Sets, *Computer, Communications and Electronics (Comptelix), 2017 International Conference on IEEE*, 2017.

[41] K. Qin and S. Jing, The Attribute Reductions Based on Indiscernibility and Discernibility Relations, *In International Joint Conference on Rough Sets*, Cham, Springer, 2017, pp. 306–316.

[42] K.A. Vidhya and T.V. Geetha, Resolving Entity in a Large Scale - Determining Linked Entities and Grouping Similar Attributes Represented in Assorted Terminologies, Distributed and Parallel Databases, 2017. https://doi.org/10.1007/s10619-017-7205-1.