ENTITY RESOLUTION AND BLOCKING: A REVIEW

K.A.Vidhya
Research Scholar, Dept of CSE
Anna University
Chennai, India
avidhya06@gmail.com

T.V.Geetha
Senior Professor, Dept of CSE
Anna University
Chennai, India
tv_g@hotmail.com

***Abstract* - Entity Resolution (ER) is a prerequisite to several Web applications including enhancing semantic searches and information extraction from the Web, strengthening the Web of Data by interlinking entity descriptions from autonomous sources, and supporting reasoning using related ontologies. While designing an ER system, it is assumed that each entity profile contains of a uniquely identified set of attribute-value pairs, each entity profile matches to a solitary real-world object, and two similar profiles are identified as long as they co-occur in at least one block. ER is an inherently quadratic problem (i.e., O (n2)), given that every entity must draw a comparison with others. ER does not scale to large entity collections, as in Web data. The most well-known solution for addressing large-scale ER in the literature is blocking, which is an approximate solution where similar entities are grouped into blocks and comparisons are limited to within blocks. The process of entity resolution and the types of entity resolution in relational and Web data are discussed in this paper. Further, the paper reviews the literature on the approaches introduced by former researchers on the entity resolution system. The data integration, block building, and block processing phases, and the challenges involved for designing an efficient ER system are discussed. This paper concludes with the measures required to evaluate entity resolution approaches.**

Index Terms – Entity Resolution, Linked open data, Rough Set, Web Of Data.

I. INTRODUCTION

A number of Entity Resolution (ER) systems have been proposed in the literature to enable interoperability in exponentially growing, heterogeneous Web data. ER is a complex process that is increasingly required as the size of LOD increases. Hence, diversified techniques have been introduced to make the resolution process efficient and scalable [1]. Reducing the number of comparisons is possible using three techniques: blocking, learning, and iterative. In the blocking technique, two entity profiles are matched using the blocking key value [1, 2], to block similar entity pairs. Only similar entity pairs discovered in the block building process are matched using block processing methods. The blocking is either partitioning-based, where each entity profile is kept in a single block or overlapping-based, where the EP is placed in multiple blocks [1]. First, entities are partitioned into blocks (clusters) using a partition-based blocking technique. Finally, only similar cluster blocks are matched, greatly reducing the number of comparisons. A smaller number of comparisons culminates in reduced time complexity as the number of matching computations is reduced [1, 3]. Distributed processing is a candid approach that reduces the execution time of the matching and so increases the efficiency of the system [4,5,6,], where blocking is executed either in parallel or there is block processing in parallel.

The automatic selection and fine-tuning of either of the approaches is carried out, based on the characteristics of the input LOD. In the learning-based approach, the model built will automatically learn patterns to find similar real-world objects [1]. In the iterative-based approach, the matches will progressively match similar entities from an initial set of candidate entities [2, 1]. According to Papadakis et al., [7] a blocking-based ER system is one of the best techniques for efficient matching. Because large blocks are the reason for the high time and memory-consuming resolution process, they are handled by the partitioning technique. A few systems have tried to prune entity pairs to reduce the time required by the block processing technique, but the pruning would lead to the loss of potentially similar entities. Hence, we have chosen the best and most efficient blocking technique, and attempted to make it even better by integrating it with other effective techniques. Existing work related to each technique is discussed in detail in the following subsections, with the mathematical notations for ER. As data on the Web can be accessed in miscellaneous forms, those forms have different degrees of structuredness that constrain entity resolution methods. Different types of ER use different types of blocking approaches and extend the process to enhance metrics related to the ER System. Types of ER methods and their working are discussed in the next section.

II. TYPES OF ER

As discussed in Introduction ER types are elaborated on in this section. The ER system is designed to work with iterative, blocking or learning algorithms. A detailed study of existing ER approaches in the literature for resolving entities [8,1] is presented in the next section. Variations in the blocking methods are discussed in the next few sections.

A. Pairwise ER

Pairwise ER techniques compare only two entity profiles at a time, based on pairwise similarities, to decide whether or not two records are matched [1,3]. Pairwise matching based ER relies extensively on machine learning approaches to measure the score of the individual feature vector and decide the threshold for

segregating matched records from non-matches. Existing research employs techniques such as the SVM (Support Vector Machine), decision trees, conditional random field, and an ensemble of classifiers under the machine learning techniques umbrella to resolve problems with pairwise ER. Apart from learning methods, researchers [9] proposed a bipartite method for computing pair-wise ER.

However, the efficiency of machine learning algorithms relies greatly on the availability of training sets, which are expensive; and the fact that the class must have an appropriate balance of positive and negative samples while training [1]. Later researchers resorted to crowdsourcing to create training sets or minimize the number of training samples so as to do away with supervised training. Alternative techniques used for supervised ER learning are Expectation Maximization, semi-supervised techniques, and generative models. Apart from learning methods, researchers Gupta *et al.* [9] proposed a bipartite method to compute pairwise ER.

Supervised machine learning approach called Active Learning queries interactively the information sources to get the desired output. Active learning-based ER approaches focus on optimizing precision and recall [10, 11], and the work by Jiannan Wang *et al.* [12] discussed the process, using crowdsourcing for selective labeling. It is based on pairwise classification while determining which entities are to be merged into categories. However, threshold-based pairwise entity resolution fails to address issues related to noisy data while considering records of learning pairs. Hence, recent research has extensively studied collective entity resolution methods to resolve entities accurately.

B. Collective ER

Collective ER techniques compare a set of related records that use cluster metrics for matching entities. The matching entities are grouped by various clustering algorithms, namely i) correlation clustering, ii) hierarchical clustering, and iii) nearest neighborhood-based clustering. Clustering techniques require a canonical entity or cluster representative to identify clusters with the most information. Collective ER stores a collection of records based on similar records which potentially refer to the same entity in the same cluster. For example, in ACM paper citations, the records consist of attributes such as authors, paper, venue, and conference. In this example, to recognize two author-related attributes, it is necessary to have the same author presented in the same conferences with an identical paper title. Thus, collective ER provides potential benefits that ensure accurate results by only exploiting attribute similarity measures of relational evidence. Collective ER approaches employ several models for a cluster membership decision, such as non-probabilistic similarity propagation and probabilistic similarity propagation, as undirected and generative models, alongside hybrid approaches [10].

Traditional collective entity resolution often interprets the dependency between matched sets of entities. Given the heterogeneous nature of Web data, a relational clustering algorithm [13] employs relation and attribute-level information for similar types of entities. The proposed algorithm performs collective entity resolution for two different types of entities, including the protein-protein sequence and author-publication networks. Mesh [14] is a factor-graph model which resolves entities of personal profiles based on graph-based disambiguation using Linked Data. The unsupervised entity resolution approach [15] emphasiz**es** the advantage of indexing or blocking techniques to enhance the scalability of the ER algorithm using a multi-type graph model. However, with the evolving nature of heterogeneous Web data, an iterative ER approach is called for, as discussed in the next subsection.

C. Iterative ER

In addressing large-scale Web data, the iterative approach tackles ER by generating a set of seed patterns and subsequently improvising or revising the previous matching decision. Iterative blocking can be implemented where there is a need for quick solutions that run iteratively in order to fetch the most exact matches. There are two ways of generating candidate pairs: merge-based iterative ER [16] and relationship-based iterative ER. In merge-based **iterative ER,** the matching decision among entity profiles generates a merge operation which modifies the initial entity profile with a new, merged entity profile. An example is the Sorted Neighborhood method which creates partitions and undertakes comparisons only within a partition [17]. The algorithm distributes the process of matching and merging across multiple processors. Furthermore, properties like idempotence, commutativity and associativity have been incorporated to improve efficiency. HARRA [18] extended iterative blocking by employing LSH (Locality Sensitive Hashing), which hashes input entity profiles into different buckets corresponding to the blocks in reusable hash tables. Benjelloun *et al.* (2009) proposed an R-swoosh and a hierarchical agglomerative clustering (HAC) approach

[19] where, at any stage, a cluster of descriptions implies that the entity profiles in the clusters match. In the latter method of iterative ER, duplicates are identified using the relationship established. The transitive relation, duplicates, and merge-dependency properties are identified for resolving entities. All these methods improve effectiveness by identifying superfluous duplicate matches. The next subsection discusses yet another ER approach called Incremental ER.

D. Incremental ER

Whang *et al.* [17] proposed an incremental ER framework which works in three steps, steps. The first step is blocking, the second is computing pairwise similarity within the blocks, and the third is constructing a graph where each node represents an entity and the edges have the weight of the similarity measure between the entities. This exploits the polynomial time approximation for correlational clustering and produces good results. Correlation clustering is applied to a set of descriptions and a new ER result is achieved after replacing the previous cluster of descriptions. This is an NP-complete problem and hence optimal algorithms are infeasible. The authors subsequently proposed a polynomial time algorithm that yielded better results. Similarly, Welch, Michael J *et al.* [20] used ER results to incrementally resolve entities providing real-time queries. Whang & Garcia-Molina [21] proposed a method of matching rules between entity profiles that automatically evolve. Another ER approach, termed progressive ER and outlined in the next subsection, maximizes results by taking advantage of partial results.

E. Progressive ER

Recent advances in research have veered towards progressive ER, which maximizes the reported matches given a limited number of comparisons, by taking advantage of the partial matched results obtained. Progressive ER extends the workflow by introducing a scheduling phase that randomly selects results from the blocking phase and matches entities in the entity matching phase. The optimal results are selected and again updated to the scheduling phase, and the entity matching progressively done using the results. Whang *et al.* [22] proposed three different heuristics for modeling the scheduling phase. Papenbrock *et al.* [23] introduced a progressive sorted neighborhood method which extends the first heuristics of Whang *et al.* [22] to capture matching pairs in dense areas. Finally, the meta-blocking approach proposed by Papadakis *et al.* (2014) as a progressive ER method works on the heuristics of selecting the top-k edges of a blocking graph.

Depending upon the descending order of their weights, edges are stored. Summarizing the inherent distribution of entity profiles in different KBs, alongside their significant semantic and structural diversities yields incomplete information in terms of the similarity of entity profiles published in Web data. Typical ER approaches assume that distinct blocking and matching phases may reduce the number of missed matches in the LOD cloud. Existing iterative approaches integrate ER with blocking, merging the two. It does not address dirty ER. This calls for an efficient way of addressing ER. An appropriate similarity measure is to be chosen to compare entities. A survey of existing similarity measures in the literature for the design of an efficient ER is presented below.

III. Similarity Measures for Matching Entity Profiles

The aim of an ER system is to find a set of 'M' matches that represent the same real-world entities. The total number of matched entities is determined by a set, S. The similarity measure of the matching system tries to yield a maximum value, but might not work for all types of data. The precision and recall of the similarity measure are given by $|M \cap S|/|S|$ and $|M \cap S|/|M|$ respectively, where S is the suggested match. In this section, a survey of the major similarity measures employed to resolve entity profiles in the Web of Data has been presented. Entities are compared in terms of content (attributes) or structural relations with other entity profiles. The next section discusses attribute-related similarity measures.

A. Attribute-Based Similarity Measure

Attribute-based similarity measures take as input a pair of strings, and are very useful when comparing the values of a fixed set of attributes in homogeneous data with pre-defined schema. However, the assumption is not true in the case of Web data when entity profiles are loosely structured, as discussed in literature. In order to tackle the issue of varying attributes, researchers used a token-based similarity measure, operating on the values of the entity profile so that it splits as a set of tokens or n-grams. Similarity measures like the Jaccard, Dice and cosine similarity for two sets P and Q, given in equation 1.1 , are used to find the similarity between tokens (attribute→value). The more similar the tokens, the greater the similarity between the entities. Overlap similarity takes the number of tokens in common, and divides it by the number of tokens in a smaller set.

$$\text{Jaccard } (P,Q) = \frac{P \cup Q}{P \cap Q} \qquad \text{Overlap } (P,Q) = \frac{P \cup Q}{\min(|P|,|Q|)}$$

$$\text{Dice } (P,Q) = \frac{P \cup Q}{|P|+|Q|} \qquad \text{Cosine } (\vec{P},\vec{Q}) = \frac{\vec{P}.\vec{Q}}{||\vec{P}||\,||\vec{Q}||} \qquad (1.1)$$

Cosine similarity uses features vectors calculated using the TF-IDF model and compares the two entity profiles, taking

into account tokens of a reduced frequency count. The other similarity measure, used in information theory, can quantify the statistical similarity between two entity profiles and their interdependency. The method uses the statistical inference of attributes and values and their co-occurrence. Using Formula 1.5, the mutual dependence of two random variables, 'm' and 'n', is measured by the MI (Mutual Information). The coherency score of the schema is measured by the Pointwise Mutual Information (PMI), using Formula 1.3.

$$MI(M;N) = \sum_{n\epsilon N}\sum_{m\epsilon M} p(m,n)\ln\left(\frac{p(m,n)}{p(m).p(n)}\right) \quad (1.2)$$

$$PMI(m,n) = \ln\left(\frac{p(m,n)}{p(m).p(n)}\right) \quad (1.3)$$

The PMI works better for Web table representations of Web data. All the similarity measures only discuss the compared values of entity profiles. In the next section, we discuss similarity measures devised to address attribute relationships.

B. *Relation-Based Similarity Measures*

Relational similarity measures consider neighboring entities and ascertain if they are linked. The relational similarity measure is either tree-based or graph based, and expressed in terms of a neighborhood similarity of the entity profile and a linear combination of the similarity value. According to a research by Ananthakrishna et al. [24], the DELPHI containment metric is a similarity value which considers both similarity of the related children sets and attribute similarity. The tuples are divided into token sets, 'ST', and the Levenshtein distance of the token is computed using the IDF (Inverse Document Frequency) measure. The similarity measure of one tuple 'T', covered by another tuple' T', is given by equation (1.4),

$$\mathrm{simTok}(T,T') = \frac{\sum \mathrm{idf}(ST(T)\cap ST(T'))}{\sum \mathrm{idf}(ST(T))} \quad (1.4)$$

Whereas the similarity measure for the children set is determined by equation 1.5. 'T' is the tuple set covered by the children set 'T', where CS is the children similarity measure. Both similarity measures are assigned an Inverse Document Frequency (IDF) weight. A classification function has been designed, and if the final result equals 1, then both the tuples are duplicate, otherwise they are not.

$$\mathrm{CsimTok}(T,T') = \frac{\sum |(CS(T)\cap CS(T')|}{\sum |CS(T)|} \quad (1.5)$$

The algorithm works as a top-down and bottom-up approach. When compared to the tree-based approach, graph-structured data is more complicated in terms of calculating the similarity. Bohm et al. [25] proposed LINDA, a similarity measure to address RDF triples that are only URIs. Let an assignment matrix, X, be a square binary Ɛ * Ɛ matrix, representing entity descriptions for already identified matches. Given G, X, the similarity of the system is calculated using the Lprior and Lcontext. Lprior represents literal values and Lcontext refers to the similarity function of the neighbor's ei and ej

$$\mathrm{LINDAsim} = \mathrm{Lprioir}\left(V(e_i), V(e_j)\right) + \propto \mathrm{Lcontext}\left(C(e_i), C(e_j), G, X\right) - \theta \quad (1.6)$$

where $\propto$ refers to the empirically-tuned weight, θ replaces the normalization factor of the graph-based similarity measure, and e_i and e_j are the two entity profiles. The measure is further modified into the upper and lower similarity measures of the given RDF graph. The next subsection discusses the approximation-based similarity measure in detail.

C. *Approximation of the Similarity Measure*

In the resolution of large-scale datasets of entity profiles in Web data, there are two key data processing issues: (a) to shorten the representation of entity descriptions in the main memory, and (b) to do away with systematically comparing all pairs of descriptions for similarity. To address these issues, a large set of tokens has been used for conversion into short signatures called key values. In this regard, we present, briefly, the ideas underlying these approximation techniques to resolve entities. Min-hashing is a technique for tackling high-dimensional text comparisons, originally designed for computing the similarity of documents, represented as small signatures rather than a large sequence of k-shingles. To find the similarity between entity descriptions, it employs a signature matrix and permutation measures to calculate the similarity between entities.

The min-hash signature fits into the main memory for comparison, though comparing all the columns culminates in quadratic complexity. Locality Sensitive Hashing (LSH) [27] relies on an adequate function that establishes whether or not functions x and y constitute a candidate pair. To conclude, an effective ER system demands crucial contextual information, apart from the similarity measure. To improve the efficiency of the ER system so as to reduce quadratic comparisons, LSH or blocking is performed. Both methods aim for a trade-off in the number of missed matches and discarded non-matches. However, when applied across domains, the LSH method has difficulty tuning the similarity threshold. The blocking technique, on the contrary, seems more resilient at addressing the diversity and incompleteness of Web data. Similarity measures and types associated with ER are discussed below. The next section details entity resolution frameworks and a comparative study of the frameworks is made.

IV. Entity Resolution Frameworks

In entity resolution for Web data [1], an entity refers to real-world objects present in news articles, home pages, blog entries, microposts, comments, and websites. Entity resolution is crucial to data cleaning and integration. With exponential increases in Web content updation, the Web has become one of the world's largest repositories. With plenty of Web data at hand, mostly in the form of natural language, there is every possibility of the occurrence of highly ambiguous data, especially named entities.

Table 1.1 Comparison of Matching Frameworks

Strategy		Framework	Entity Type	Partitioning	Matchers	Learners
Entity Matching with training		Multiple Adaptive Record Linkage with Induction (MARLIN) (Bilenko Mikhail and Raymond J. Mooney) [28]	Relational	Canopy clustering	Attribute value	SVM and Decision tree
		Operator Trees (Chaudhuri Surajit *et al.*) [29]	Relational	Canopy clustering	Attribute value	SVM and Operator tree algorithms
		Active Atlas (Tejada, Sheila *et al.*) [30]	Relational	Hashing	Attribute value	Decision tree
Entity Matching without training		Bayesian Network (BN) (Leitão, Luís *et al.*) [31]	Extensible Markup Language (XML)	----	Attribute value and context	----
		Mapping-based Object Matching (MOMA) (Thor Andreas and Erhard Rahm) [32]	Relational	----	Attribute value and context	----
		Stanford Entity Resolution Framework (SERF) (Benjelloun Omar *et al.*) [19]	Relational	----	Attribute value	----
Strategy		Framework	Entity Type	Partitioning	Matchers	Learners
Hybrid Entity Matching Systems		TAILOR (Elfeky, Mohamed *et al.)* [35]	Relational	Sorted neighborhood	Attribute value	Probabilistic and Decision tree
		Freely Extensible Biomedical Record Linkage (FEBRL) (Christen Peter) [33]	Relational	Sorted neighborhood, Canopy clustering, and Q-gram	Attribute value	SVM
		Self-Tuning Entity Matching (STEM) (Köpcke Hanna and Erhard Rahm) [34]	Relational	Sorted neighborhood	Attribute value	SVM, Decision tree, Multiple learning, and Logistic regression
Hybrid Entity Matching Systems		Context-based Framework (Chen Zhaoqi *et al.*) [36]	Relational	Canopy-like	Attribute value and context	SMOreg and Logistic regression

ER is critical to dynamically updating such information on the Web. Though several existing tools reliably recognize similarly-named entities in Web documents, the extracted entities are non-unique, since even the name does not belong to the same entity in different Web pages. Later, researchers focused on handling this as a disambiguation problem that is very similar to the ER problem. A number of early techniques, available to resolve ER problems in citation domains and datasets of semi-structured data, is presented below.

These traditional ER techniques are not readily applicable for resolving entities in Web document collections, given the challenges to be reckoned with, especially in terms of unstructured data. Real-time ER is essential for many applications. For example, credit bureaus require a real-time ER model on dynamic datasets to undertake processes in real time. The bank sends customers' details to the credit bureau to check focussed on handling this as a disambiguation problem that is very similar to the ER problem.

A number of early techniques, available to resolve ER problems in citation domains and datasets of semi-structured data, is presented below. The credit score when a person applies for a loan from the bank. In this case, the credit bureau has to link details of customers with their database and send the corresponding credit history to the bank.

The decision of loan approval by the bank ultimately relies on the credit history received from the credit bureau in which real-time ER helps provide instant responses to the bank. To handle the rapid growth of dynamic real-world data, traditional ER approaches conduct efficient query matching. The steps involved in Web or real-time ER are similar to the traditional ER process, and include data cleaning and standardization, indexing, comparison, and classification. Table 1.1 discusses different entity matching frameworks and their characteristics like the partitioning, matching or learning techniques used in existing work. Existing ER techniques and pairwise or cluster-based similarity measures have been discussed.

However, heterogeneous Web data are prone to heterogeneity at the attribute and schema levels, and uncertainty in Web data is to be resolved. Rough Set Theory is a mathematical model that addresses uncertainty in data, missing values and feature selection in large dimensional data. Consequently, a survey of rough set theory concepts and its contributions to clustering is carried out by Vidhya & Geetha, [37] in order to prove the importance of Rough Set Theory in with exact results that also appear in the predicted precision is defined as the percentage of matches in the predicted results that are relevant. Pairwise recall is defined as the percentage of matches results.

V. Entity Resolution Evaluation Measures

The evaluation of entity resolution algorithms involves the prediction of correctness, compared to previously annotated ground-truth information. Most notably, several measures have been increasingly used to evaluating the ER system that is categorized into three main categories: pairwise, cluster, and edit distances. Moreover, it is essential to evaluate the performance of indexing techniques with real-time ER. Accordingly, evaluating the effectiveness, efficiency, and scalability of the ER process is necessary. In the ER system, the evaluation of block building methods needs to maintain a balance between the Pair Quality (PQ), Pair Completeness (PC), and Reduction Ratio (RR) evaluation.

A. Pairwise Measures

Pairwise evaluation measures compute all possible pairs of entities or records to evaluate the ER system in which each pair refers to the link between two references, entities, or records. Evaluation metrics such as precision and recall belong to pairwise measures. Moreover, the harmonic mean of these evaluation metrics produces the most frequently used metric in ER. Pairwise evaluation metrics assist in assessing an intuitive interpretation of the matches. Pairwise

B. *Cluster Measures*

Cluster evaluation measures compare clusters or blocks to evaluate ER results in which the blocks consist of linked entities or records. To estimate the cluster performance, the resulting cluster is compared with ground-truth clusters. This kind of evaluation attempts a holistic understanding of the resolved entities. Cluster evaluation measures involve the cluster F1, the closest cluster F1, Message Understanding Conference (MUC) F1, B3 F1, and Constrained Entity-Alignment F-Measure (CEAF). Of these evaluation metrics, Cluster F1 and the closest cluster F1 metrics are discussed as follows.

Cluster precision is the ratio between the number of entirely correct clusters and the total number of retrieved clusters in the result. Cluster recall is the ratio between the number of completely correct clusters and the total number of exact clusters in the ground truth. In contrast to pairwise metrics, cluster level precision and recall metrics consider exact matches in terms of clusters. Corrupted matches in a cluster negatively impact the entire cluster results because of precise comparison-based cluster evaluation measures. The advantage of the cluster evaluation measure lies in its assessment of the significance of the entire cluster and its correctness rather than providing high scores to partially correct clusters by way of checking co-references in the clusters.

While evaluating the matching quality of the ER process, evaluation measures deal with two types of errors: False Positives (FP) and False Negatives (FN). FP refers to incorrectly classified record pairs as matches that are irrelevant to actual true matches. FN defines incorrectly classified record pairs as non-matches that are relevant to actual true matches.

Moreover, True Positives (TP) refer to correctly classified record pairs as matches which are relevant to the actual true matches. True Negatives (TN) are classified record pairs of non-matches that are relevant to actual non-matches. According to these error types, ER approaches evaluate the quality of the classified record pairs through precision and recall.

Mean Reciprocal Rank (MRR): MMR is an overall relevant performance measure that validates the matching process. It is the average of the reciprocal rank of the true matching record in sample query records in a query stream.

Efficiency and Scalability: To evaluate the efficiency and scalability of the ER approach, time-related evaluation measures have been used twice. The scalability of the system depends purely on the evaluation of both the average insertion and query time with growing indexes over large-scale datasets. Average insertion time is the average time taken to insert a query record into the index database. Average query time is defined as the average time taken to match a query with the records in the index database. Having conducted an exhaustive literature survey, the following section lists the overall challenges addressed in the thesis.

VI. *Overall Challenges*

The general challenges in existing ER systems are detailed below. Existing relational ER systems are unsuitable for Web data, given the high heterogeneity (even across domains) and non-regularity in data structuring. A major challenge addressed in this thesis is loose schema binding. LOD is semi-structured data that is loosely bound to a rich diversity of schemata ranging from locally-defined attribute names at an unprecedented level of heterogeneity. This pertains both to the schema describing the same entity types, and also to separate profiles describing the same entity. Consequently, handling the complexity of heterogeneous Web data by removing duplicates and linking entities becomes a challenge. This is addressed by resolving attribute and schema-level heterogeneity and designing three novel blocking methods. The next challenge addressed in this thesis is the high levels of noisy data. This is because Web data are published through free resources, and existing approaches are unable to filter low-quality information. Such unfiltered low-quality information abounds in noise ranging from spelling mistakes to missing information and inconsistent values. False information can hamper the identification of matching entities, thus creating blocks with low efficiency.

In existing blocking approaches, the efficiency of the blocking algorithm can be improved while reducing the number of pairwise comparisons. Despite the various indexing methods in blocking, the number of pairwise comparisons needs to be reduced further. Based on the review, it can be concluded that reducing the number of comparisons with improved efficiency and effectiveness is crucial, owing to the nature of fast-evolving Web data. Another challenge addressed in this thesis is the data integration of Web data. Given the voluminous, evolving nature of Web data, an efficient ER system is required to handle both relational and attribute-level information across Web data. This necessitates an efficient collective ER approach and a MapReduce framework to address scalability in large-scale ER.

Real-time query processing is yet another challenge that has to be addressed. Given a query, retrieving similar real-world entities from heterogeneous Web data with a reduced response time is vital. The overall challenges are listed, and in the following chapters we have addressed these challenges accordingly to improve the efficiency and effectiveness of the blocking approach for an ER system. The literature survey is concluded below.

VII. *Conclusion*

The vast variety of existing blocking methods and their open challenges have been discussed. Specifically, efficient blocking measures such as the distance-based and similarity-based measures have been outlined. In existing blocking approaches, the efficiency of the blocking algorithm can be improved with a reduced number of pairwise comparisons. Despite the various indexing methods in blocking, the number of pairwise comparisons still needs to be reduced. Further, the effectiveness of the supervised algorithms for the blocking methodology with limited training data was unsatisfactory. Finally, various evaluation metrics with respect to the blocking algorithms are outlined.

REFERENCES

[1] Vasilis, Kostas Stefanidis & Vassilis Christophides 2015, ‘Big data entity resolution: From highly to somehow similar entity descriptions in the Web’, IEEE International Conference on Big Data (Big Data), pp. 401-410.

[2] Christen Peter 2012, ‘A survey of indexing techniques for scalable record linkage and deduplication’, IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 9, pp. 1537-1555.

[3] Papadakis George, Ekaterini Ioannou, Claudia Niederée & Peter Fankhauser 2011a, ‘Efficient entity resolution for large heterogeneous information spaces’, ACM Proceedings of the Fourth International Conference on Web Search and Data Mining, pp. 535-544.

[4] Gruetze, Toni, Christoph Böhm & Felix Naumann 2012, ‘Holistic and Scalable Ontology Alignment for Linked Open Data’, LDOW 937.

[5] Parundekar, Rahul, Craig A Knoblock & José Luis Ambite 2013, ‘Discovering Alignments in Ontologies of Linked Data’, In IJCAI, pp. 3032-3036.

[6] Fisher Jeffrey, Peter Christen, Qing Wang & Erhard Rahm 2015, ‘A clustering-based framework to control block sizes for entity resolution’, 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 279-288.

[7] Papadakis George, Ekaterini Ioannou, Themis Palpanas, Claudia Niederee & Wolfgang Nejdl 2013, ‘A blocking framework for entity resolution in highly heterogeneous information spaces’, IEEE Transaction on Knowledge Data Engineering, vol. 25, no. 12, pp. 2665-2682.

[8] Shen Wei, Jianyong Wang & Jiawei Han, 2015, ‘Entity linking with a knowledge base: Issues, techniques, and solutions’, IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 2, pp. 443-460.

[9] Gupta, Rahul & Sunita Sarawagi 2009, ‘Answering table augmentation queries from unstructured lists on the web’, Proceedings of the VLDB Endowment 2.1, pp. 289-300.

[10] Arvind Arasu, Michaela Gotz & Raghav Kaushik, 2010, 'On active learning of record matching packages', In Proceedings of ACM SIGMOD International Conference on Management of data, SIGMOD'10, pp. 783-794.
[11] Kedar Bellare, Suresh Iyengar, Aditya G. Parameswaran & Vibhor Rastogi 2012, 'Active sampling for entity matching', In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'12, pp. 1131-1139.
[12] Jiannan Wang, Tim Kraska, Michael J. Franklin & Jianhua Feng 2012, 'Crowder: crowdsourcing entity resolution', Proceedings VLDB Endow, vol. 5, no.11, pp. 1483-1494.
[13] Bhattacharya Indrajit & Lise Getoor, 2007, 'Collective entity resolution in relational data', ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 1, no.1.
[14] Cudré-Mauroux, Philippe, Parisa Haghani, Michael Jost, Karl Aberer, & Hermann De Meer 2009, 'idMesh: graph-based disambiguation of linked data', ACM Proceedings of the 18th International Conference on World Wide Web, pp. 591-600.
[15] Zhu Linhong, Majid Ghasemi-Gol, Pedro Szekely, Aram Galstyan & Craig A. Knoblock 2016, 'Unsupervised Entity Resolution on Multi-type Graphs', In International Semantic Web Conference, Springer International Publishing, pp. 649-667.
[16] Whang Steven Euijong, David Menestrina, Georgia Koutrika, Martin Theobald, and Hector Garcia-Molina, 2009, 'Entity resolution with iterative blocking', ACM Proceedings of the SIGMOD International Conference on Management of data, pp. 219-232.
[17] Hernández, Mauricio A & Salvatore J. Stolfo 1995, 'The merge/purge problem for large databases', In ACM Sigmod Record, vol. 24, no. 2, pp. 127-138.
[18] Kim Hung-Sik & Dongwon Lee, 2010, 'HARRA: fast iterative hashed record linkage for large-scale data collections', ACM Proceedings of the 13th International Conference on Extending Database Technology, pp. 525-536.
[19] Benjelloun Omar, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang & Jennifer Widom, 2009, 'Swoosh: a generic approach to entity resolution', The VLDB Journal - The International Journal on Very Large Data Bases, vol. 18, no. 1, pp. 255-276.
[20] Welch, Michael J, Aamod Sane & Chris Drome 2012, 'Fast and accurate incremental entity resolution relative to an entity knowledge base.' In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 2667-2670.
[21] Whang Steven Euijong, and Hector Garcia-Molina, 2014, 'Incremental entity resolution on rules and data', The VLDB Journal, vol. 23, no. 1, pp. 77-102.
[22] Whang Steven Euijong, David Marmaros & Hector Garcia-Molina, 2013, 'Pay-as-you-go entity resolution', IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, pp. 1111-1124.
[23] Papenbrock, Thorsten, Arvid Heise & Felix Naumann 2015, 'Progressive duplicate detection.' IEEE Transactions on knowledge and data engineering, vol. 27, no. 5, pp. 1316-1329.
[24] Ananthakrishna, R, Chaudhuri, S, & Ganti, V 2002, 'Eliminating fuzzy duplicates in data warehouses', In VLDB'02: Proceedings of the 28th International Conference on Very Large Databases, pp. 586-597.
[25] Böhm, Christoph, Gerard de Melo, Felix Naumann, and Gerhard Weikum 2012, 'LINDA: distributed web-of-data-scale entity matching', In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 2104-2108. ACM.
[26] Leskovec, Jure, Anand Rajaraman & Jeffrey David Ullman 2014, 'Mining of massive datasets', Cambridge university press.
[27] Wang Qing, Mingyuan Cui & Huizhi Liang 2016, 'Semantic-aware blocking for entity resolution', IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 1, pp. 166-180.
[28] Bilenko Mikhail & Raymond J Mooney, 2003, 'On evaluation and training-set construction for duplicate detection', In Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, pp. 7-12.
[29] Chaudhuri, S, Chen, BC, Ganti, V & Kaushik, R 2007, 'Example-driven design of efficient record matching queries', In Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, pp. 327-338.
[30] Tejada Sheila, Craig A. Knoblock & Steven Minton, 2002, 'Learning domain-independent string transformation weights for high accuracy object identification', ACM Proceedings of the eighth SIGKDD international conference on Knowledge discovery and data mining, pp. 350-359.
[31] Leitão Luís, Pável Calado & Melanie Weis 2007, 'Structure-based inference of XML similarity for fuzzy duplicate detection', ACM Proceedings of the sixteenth conference on Conference on Information and Knowledge Management, pp. 293-302.
[32] Thor Andreas & Erhard Rahm, 2007, 'MOMA-A Mapping-based Object Matching System', In CIDR, pp. 247-258.
[33] Christen Peter 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, vol. 80, pp. 17-25.
[34] Köpcke Hanna & Erhard Rahm 2008, 'Training selection for tuning entity matching', In QDB/MUD, pp. 3-12.
[35] Elfeky Mohamed G, Vassilios S Verykios & Ahmed K Elmagarmid, 2002, 'Tailor: A record linkage toolbox', IEEE 18th International Conference on Data Engineering, pp. 17-28.
[36] Chen Zhaoqi, Dmitri V. Kalashnikov & Sharad Mehrotra 2009, 'Exploiting context analysis for combining multiple entity resolution systems', ACM Proceedings of the SIGMOD International Conference on Management of data, pp. 207-218.
[37] Vidhya, KA & Geetha, TV 2017, 'Rough set theory for document clustering: A review', Journal of Intelligent & Fuzzy Systems, vol. 32, no. 3, pp. 2165-2185.