Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 3, February : 2024

ENTITY MATCHING: A COMPREHENSIVE ENSEMBLE APPROACH

Dr. Vijay Maruti Shelake, Faculty, Dept. of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, University of Mumbai
Mr. Siddhesh M. Khawale, Faculty, Dept. of Computer Science, Arts, Commerce and Science, Lanja, University of Mumbai

Abstract
Entity matching is a fundamental procedure that involves pinpointing entries in separate databases that pertain to identical real-world entities. This operation is crucial for enhancing data quality and facilitating seamless data consolidation. The conventional approach involves assessing whether two record pairs match by evaluating the likeness of their respective attribute values. This comparison utilizes similarity metrics and is typically performed on selected attributes from the databases. Unfortunately, many of these methods can lead to decreased accuracy. Hence, a novel approach is introduced, suggesting the use of an ensemble strategy to compute similarity based on relevant features across the databases. This approach yields improved accuracy when analyzing various features within the proposed system.

Keywords:
Entity matching, Entity resolution, Data cleaning, Data Duplication, Data Integration

I. Introduction

Entity Matching, also interchangeably known as Entity Resolution, stands as a critical task in the realm of data integration. The task of finding similar entities in diverse data sources is called as Entity Matching. Conventional techniques for entity matching have typically emphasized either achieving comprehensive matches or optimizing efficiency. The challenge lies in addressing discrepancies such as typographical errors, misspellings, and abbreviations, which can result in different representations of the same entity. To maintain data quality for effective data mining, these divergent records must be accurately consolidated.

Entity matching presents a significant challenge when it comes to unifying diverse databases, acting as a critical link connecting the various facets of a database's entities. This operation takes two entities as input and produces a semantic correlation between their respective elements. It identifies a set of commonalities among the elements within two separate databases [1-5].

Matching a record with multiple attributes typically requires assessing the match for each attribute's values individually. Several research studies have introduced a range of string distance metrics to support this attribute value matching, with each metric specializing in addressing specific types of errors. However, these metrics predominantly function at the character or string level, and they may not effectively capture semantic similarities [6-8].

In recent research, word embedding techniques have been utilized to acquire distributed representations of records, enabling the capture of semantic similarity among attribute values [15]. However, for certain attributes like price, personal names, and location names, which lack strong semantic associations, current approaches tend to concentrate on string or character-level comparisons. It's important to recognize that different types of attributes should be treated differently. Descriptive attributes of entities, often represented by lengthy textual content, exhibit strong semantic connections. Even if these attributes suffer from misspellings or originate from diverse data sources, they maintain similar semantics. Semantic-level similarity is suitable for this category of attributes. Conversely, for numeric or nominal attributes with weaker semantics, like price, personal names, and location names, string-level or character-level comparisons are typically preferred, as indicated by references [11],[4],[18],[20],[10].

The process of entity matching and value mapping between two separate and dissimilar information sources plays a crucial role in applications related to data cleaning and integration, data warehousing, and database federation. Before consolidating data from multiple tables, it is essential to ensure the alignment of columns and values across these tables [14],[17].

In many instances, existing entity matching methods fail to fully harness the wealth of available information concerning entities during the calculation of similarity. This accuracy of the system can be directly impacted by the limitation. In response to this challenge, the system has introduced an innovative approach to integrate diverse entities through entity matching. This approach leverages various features within the databases to enhance the matching process and improve overall accuracy.

II. Literature Review

Entity matching was initially proposed by Newcombe under the name "record linkage" in order to identify medical records belonging to the same patient that span multiple time periods. Subsequently, Entity Matching has been investigated under a multitude of designations, including merge/purge, duplicate detection, entity resolution, and object identification, among others. Research in this field can be broadly categorized into machine learning-based, rule-based, and crowd-based approaches.

Notably, Fellegi I.J. and Sunter A.B. laid the foundation for formal theory and statistical methods for Entity Matching, and many subsequent machine learning techniques have built upon their work [11]. A prevalent approach in machine learning conceptualizes entity matching as a binary classification task, where "no-match" and "match" represent the two classes. Generally, this is accomplished by determining the degree of similarity between attributes that correspond to two entities. A classifier is then trained using this similarity vector, or a threshold is established to compare the similarity of each attribute against this threshold, determining whether the two entities constitute a match.

As an example, references [18] and [10] examine the similarity of "compatible" neighbors linked through pre-aligned relations, while [20] focuses on neighbors connected through relations with similar labels. Notably, their approach doesn't combine various similarities into a single score but rather employs a disjunction of different pieces of evidence drawn from values, neighbors, and descriptions. The identification of the most significant neighbors is an automated process based on dataset statistics. In terms of matching decisions, Entity Resolution (ER) can be categorized as pairwise or collective. In pairwise ER, matching decisions depend solely on the similarity of attribute values within descriptions. In contrast, collective ER updates matching decisions iteratively by dynamically assessing the similarity of entity neighbors.

A static collective approach is suggested by the authors, in which all sources of similarity are assessed singularly for each candidate combination throughout a predetermined number of iterations. In particular, the methodology described in reference [18] commences by selecting seed matches that contain identical entity names. Subsequently, matching decisions are expanded to encompass compatible neighbours of the initial matches. By utilizing Unique Mapping Clustering, connections are identified. In the beginning, all pairings are arranged in a priority queue in descending similarity. In every iteration, the top combination is deemed to be a match if their similarity surpasses a predetermined threshold denoted as "t" and none of their entities have been matched previously. The relevance of neighbouring similarities is reassessed for every newly matched pair, resulting in an update to the priority queue positions of the pairs. Continue until the similarity of the leading pair is reduced to a level below the predetermined threshold "t."

On the other hand, [20] considers compatible neighbours connected via relations with similar names, a condition rarely encountered in the Web of Data, which is a novel approach. [10], meanwhile, employs a comparable methodology but incorporates a number of significant alterations. Initially, a set of blocks is traversed by the matching procedure, as opposed to the initial blocks. Furthermore, it employs statistical analysis to identify unique entity names and critical relationships automatically. Furthermore, an assortment of matching evidence sources (including values, names, and neighbors) are utilized to identify candidate matches statically throughout the blocking phase.

Thus, we have reviewed the various entity matching studies by eminent researchers and it plays a crucial role in numerous real-world applications.

III. Methodology

Entity matching plays a pivotal role in the intricate process of consolidating data from diverse sources. Nonetheless, challenges arise due to the existence of errors, inconsistent data formats, and restricted data sharing across dissimilar databases, all of which hinder efforts to achieve seamless data interoperability and integration. While diverse techniques have been applied to tackle entity matching, there's an ongoing imperative to directly address data errors.

One promising approach in this regard is feature-based entity matching, which focuses on identifying matching entities within the source data. To further improve the precision of entity matching, an ensemble strategy is put forth as a solution.

In most current implementations, the entity matching process is semi-automatic due to the inherent difficulty in automatically matching different entities. This challenge stems from the fact that databases are typically designed by different individuals or teams, leading to variations in entity representation models, names, and structures.

The proposed system aims to address these issues by facilitating the matching of diverse database entities. Figure 1 illustrates two distinct databases that utilize a feature-based matching system. In this system, relevant attribute features are selected, and matching is performed based on these features. Subsequently, a global schema is derived using the feature matching strategy through similarity calculations, and the resulting data are combined based on matched features.

Figure 1: Entity Matching Strategy

The attribute type match assesses the similarity of two features' names by calculating the percentage of resemblance between them. This feature-based matching method utilizes the similarity calculation to determine how similar the names of two attributes are. The system at hand takes into account various features for entity matching. The methodology for this proposed system can be outlined as follows:

1) **Input:** The system takes data from different sources as input.

2) **Entity Matching:** The input data undergo entity matching with the aid of these features.

3**) Global Schema Development:** A global schema is constructed based on the outcomes of the entity matching process.

4)**Data Integration:** Finally, data from various sources is combined or integrated using the global schema, facilitating data interoperability and integration.

IV. Data Analysis and Findings

The implementation of the system can be carried out by following the steps below, and Python's Record Linkage toolkit can be a valuable tool for this purpose. This toolkit facilitates record linkage, helping to deduplicate records and manage data efficiently. Here are the steps for implementation:

i. Data Linkage with Python Record Linkage Toolkit: Utilize the Python Record Linkage toolkit for linking records. This toolkit is valuable for deduplicating records and effectively managing data.

Indexing for Efficiency: Employ the indexing module, which is beneficial for both linking and duplicate detection. It efficiently returns pairs from the upper triangular part of the matrix, especially when dealing with duplicate detection.

ii. Comparison with Record Linkage: Apply the capabilities of the recordlinkage library, particularly the compare class and its associated methods. This allows you to perform various types of comparisons, including string similarity measures, numerical measures, distance measures, and more, to assess the similarity between records.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 3, February : 2024

iii. Classification of Record Pairs: In this step, classify record pairs into categories such as matches, non-matches, and possible matches. You can apply both supervised and unsupervised classification algorithms to achieve this, depending on the nature of your data and matching requirements.

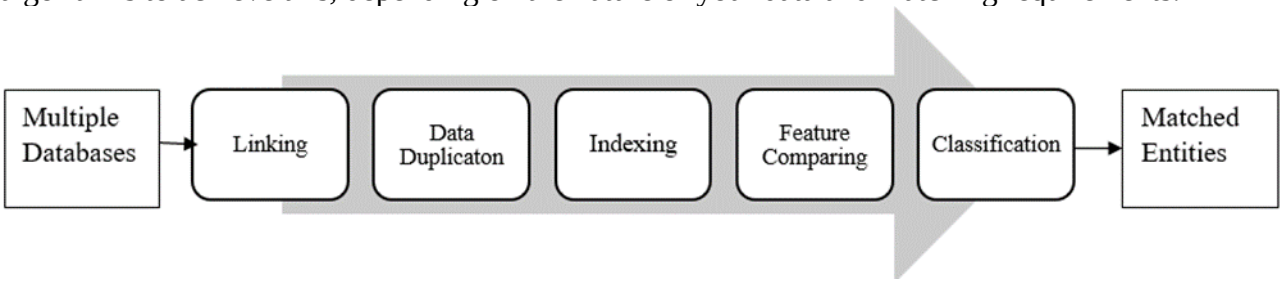

Figure 2: Implementation Steps

By following the mentioned steps and leveraging the Python Record Linkage toolkit, you can effectively implement record linkage and deduplication processes, ultimately improving the quality and management of your data. The following results were obtained by incorporating Jaro Wrinkler and Levenstein methods in the implementation steps:

A. Data Duplication:

a. Scenario I:

Figure 3 depicts the count of records discovered when employing various methods during the data duplication phase for the "surname" attribute. It's notable that the Jaro-Winkler method retrieves a greater number of records compared to the Levenshtein method.

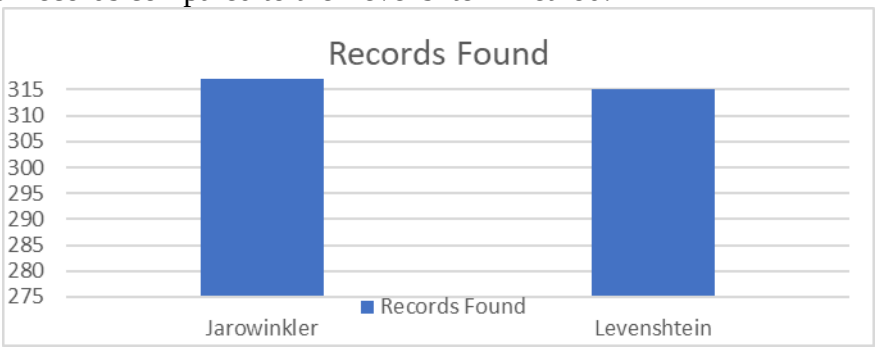

Figure 3: Graph showing records discovered in data duplication phase on surname attribute

b. Scenario II:

Figure 4 illustrates the quantity of records identified using different methods in the data duplication phase for the "address" attribute. It's evident that the Jaro-Winkler method retrieves a greater number of records in comparison to the Levenshtein method.

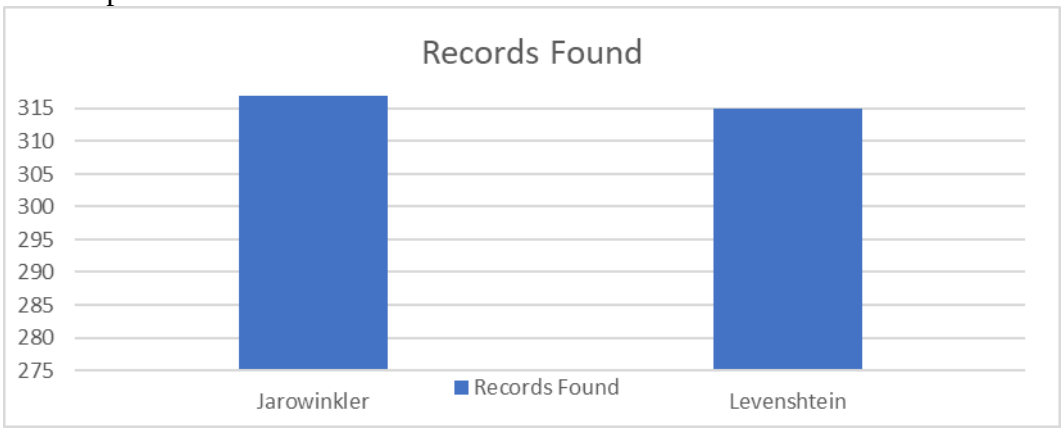

Figure 4: Graph showing records discovered in data duplication phase on address attribute

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 2, No. 3, February : 2024

B. Linking:

a. Scenario I:

In Figure 5, you can observe the number of records discovered through the methods employed during the linking phase for the "surname" attribute. Notably, the Jaro-Winkler method retrieves a greater number of records when compared to the Levenshtein method.

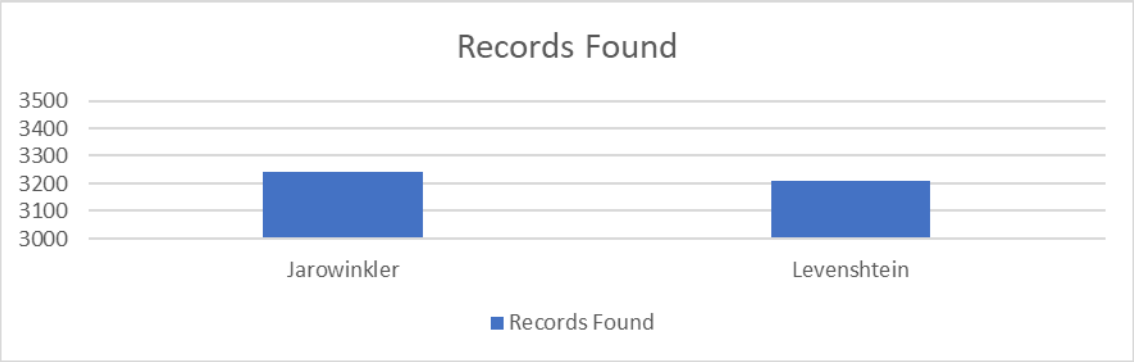

Figure 5: Graph showing records discovered in data Linking phase on surname attribute

b. Scenario II:

Figure 6 displays the quantity of records identified using the methods applied during the linking phase for the "address" attribute. It is apparent that the Jaro-Winkler method retrieves a higher number of records than the Levenshtein method.

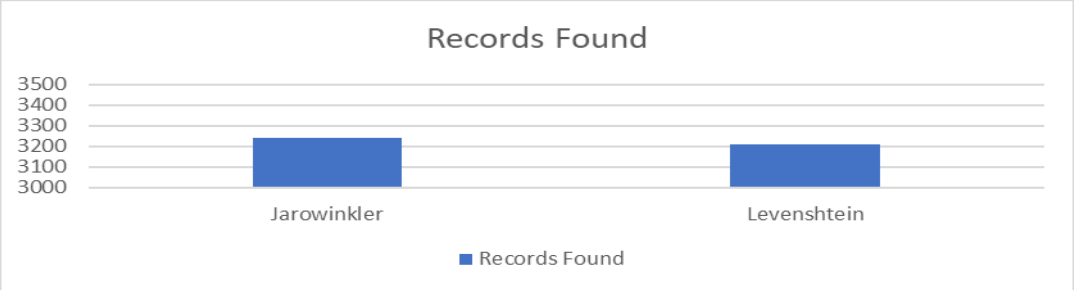

Figure 6: Graph showing records discovered in data Linking phase on surname attribute

V. Conclusion

Entity matching poses unique challenges that are specific to the characteristics of the data involved. Effective entity matching should be capable of identifying and addressing errors that may arise during the similarity calculation process. The proposed work aims to provide users with the ability to obtain matched data from multiple sources, leveraging various features for accurate matching. To enhance the matching process and ensure its effectiveness, the implementation of ensemble functionality is recommended to combine multiple matching approaches and information sources. This ensemble approach can significantly improve the quality and accuracy of matched data, contributing to more robust data integration and analysis.

This work concentrates on entity matching and the diverse methods employed in this context. With the exponential growth in data volume, the entity matching process can indeed become more intricate. The surge in duplicate records across various databases emphasizes the importance of addressing big data challenges and the need for effective duplicate removal.

Looking forward, the research focus is likely to shift toward the realm of big data and developing strategies for deduplication. Moreover, the application of these principles can be extended to real-world scenarios, including fields such as biomedical entity matching and neural network entity matching. These applications hold significant promise in improving data quality and advancing our understanding and utilization of vast and complex datasets in various domains.

References

[1] A. Bilke and F. Naumann, "Schema matching using duplicates," in Proceedings of the 21st International Conference on Data Engineering, (2005), pp. 69-80.
[2] A. Doan, P. Domingos, and A. Levy, "Learning source descriptions for data integration," in Proceedings of the 3rd International Workshop on the Web and Databases,(2000), pp. 81-86.
[3] B. T. Dai, N. Koudas, D. Srivastava, A. K. H. Tung, and S. Venkatasubramanian, "Validating multi-column schema matchings by type," in Proceedings of the 24th International Conference on Data Engineering, (2008), pp. 120-129.
[4] Bilenko M, Mooney R 1. Adaptive duplicate detection using leamable string similarity measures[C]11 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, (2003):39-48.
[5] C. E. H. Chua, R. H. L. Chiang, and E.-P. Lim, "Instance-based attribute identification in database integration," The VLDB Journal, Vol. 12, (2003), pp. 228-243.
[6] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm, "Schema and ontology matching with COMA++," in Proceedings of ACM SIGMOD International Conference on Management of Data, (2005), pp. 906-908.
[7] E. Bertino, G. Guerrini, and M. Mesiti, "A matching algorithm for measuring the structural similarity between an XML document and a DTD and its applications," Information Systems, Vol. 29,(2004), pp. 23-46.
[8] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," The VLDB Journal, Vol. 10, (2001), pp. 334-350.
[9] Ebraheem M, Thirumuruganathan S, Joty S, et al. DeepER – Deep Entity Resolution[J]. (2017).
[10] Elmagarmid, Ahmed K, Ipeirotis, Panagiotis G, Verykios, Vassilios S. Duplicate Record Detection: A Survey[J]. IEEE Transactions on Knowledge and Data Engineering, (2006), 19(1):1-16.
[11] Fellegi I P, Sunter A B. A Theory for Record Linkage[J] . Publications of the American Statistical Association, (1969), 64(328):1183-1210.
[12] J. Evermann, "Theories of meaning in schema matching: An exploratory study," Information Systems, Vol. 34, (2009), pp. 28-44.
[13] J. Kang and J. F. Naughton, "On schema matching with opaque column names and data values," in Proceedings of ACM SIGMOD International Conference on Management of Data, (2003), pp. 205-216.
[14] Khai Nguyen and Ryutaro Ichise, "ScLink: supervised instance matching system for heterogeneous repositories," National Institute of Informatics, Vol. 32, (2016), pp. 830-863
[15] Li H, Xu J. Semantic Matching in Search[J]. Foundations & Trends in Information Retrieval, (2014), 7(5):343-46.
[16] Remco de Vos, "The design and implementation of FlexiMatch", A learning, flexible & extendible framework for matching schemas, Thesis report, Enschede, May (2006).
[17] S. I. Hakak, A. Kamsin, P. Shivakumara, G. A. Gilkar, W. Z. Khan and M. Imran, "Exact String Matching Algorithms: Survey, Issues, and Future Research Directions," in IEEE Access, vol. 7, pp. 69614-69637, (2019).
[18] Singh R, Meduri V V, Elmagarmid A, et al. Synthesizing entity matching rules by examples[J]. Proceedings of the VLDB Endowment, (2017),11 (2): 189-202.
[19] W.-S. Li and C. Clifton, "SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks," Data and Knowledge Engineering, Vol. 33, (2000), pp. 49-84.
[20] Wang J, Li G, Yu J X, et al. Entity matching: how similar is similar[J], Proceedings of the Vldb Endowment, (2011), 4(10):622-633.