Improving Quality of Entity Resolution Using a Cascade Approach

Khizer Syed, Onais Khan Mohammed, John Talburt, Adeeba Tarannum,
Altaf Mohammed, Mudasar Ali Mir and Mahboob Khan Mohammed
Department of Information Quality, University of Arkansas at Little Rock, Little Rock, U.S.A.

Keywords: Entity Resolution, Record Linkage, Cascade, Group Membership Graph, Household Discovery, Entity Detection, Entity Relationship, Entity Recognition, Data Standardization, Master Data Management.

Abstract: Entity Resolution (ER) is a critical technique in data management, designed to determine whether two or more data references correspond to the same real-world entity. This process is essential for cleansing datasets and linking information across diverse records. A variant of this technique, Binary Entity Resolution, focuses on the direct comparison of data pairs without incorporating the transitive closure typically found in cluster-based approaches. Unlike cluster-based ER, where indirect linkages imply broader associations among multiple records (e.g., A is linked with B, and B is linked with C, thereby linking A with C indirectly), Binary ER performs pairwise matching, resulting in a straightforward outcome—a series of pairs from two distinct sources. In this paper, we present a novel improvement to the cascade process used in entity resolution. Specifically, our data-centric, descending confidence cascade approach systematically orders linking methods based on their confidence levels in descending order. This method ensures that higher confidence methods, which are more accurate, are applied first, potentially enhancing the accuracy of subsequent, lower-confidence methods. As a result, our approach produces better quality matches than traditional methods that do not utilize a cascading approach, leading to more accurate entity resolution while maintaining high-quality links. This improvement is particularly significant in Binary ER, where the focus is on pairwise matches, and the quality of each link is crucial.

1 INTRODUCTION

Entity Resolution techniques have evolved to help solve the problem of linking entities across large datasets and is also referred to as record linkage or deduplication. Entity Resolution is the process of identifying and matching records that pair to the same real-world entity across different data sets (Cohen, 2003). Binary Entity Resolution is an approach to evolve cluster comparisons to more effectively link pairs of records between two data sets. This technique has several advantages over traditional methods used to handle complex data structures.

The cascading process involves multiple stages where records not linked in initial attempts are passed through subsequent cascades for further analysis. This iterative approach allows the system to refine and attempt to link more records at each stage. Cascading indices can be defined as the stricter ones to match prior to the lesser strict. For example, the first cascade can be an exact match as variation i.e, 'Johnny Doe' versus 'John Doe' are considered. By the third and fourth cascades, even more lenient matching criteria might be used to link records that were previously unlinked. After completing the cascade stages, any remaining unlinked records are put through the household discovery process. Here, records are analyzed to identify households, linking records based on shared addresses or other household indicators. Links established through earlier cascades are considered strong links, while those made during household discovery are categorized as weak. (Mohammed, O.K. et al, 2024). The use of cascading stages allows the system to handle large datasets efficiently by systematically narrowing the focus of computational resources to the most promising matches first, before exploring more complex and nuanced potential links in later stages. Record linkage is the task of quickly and accurately identifying records corresponding to the same entity from one or more data sources. Entities of interest include individuals, companies, geographic regions, families, or households (Fellegi, I. P., & Sunter, 1969). Clustering divides data patterns into subsets in such a

Syed, K., Mohammed, O. K., Talburt, J., Tarannum, A., Mohammed, A., Mir, M. A. and Mohammed, M. K.
Improving Quality of Entity Resolution Using a Cascade Approach.
DOI: 10.5220/0013077400003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 60-69
ISBN: 978-989-758-737-5; ISSN: 2184-433X

way that similar patterns are clustered together. The patterns are thereby managed into a well-formed evaluation that designates the population being sampled (Winkler, W. E., 1990). Binary entity resolution (ER) and cluster-based ER represent different techniques to resolving entities in datasets. In binary ER, the focus is on comparing individual pairs of references from separate files to determine equivalence. Unlike cluster-based ER, binary ER does not use transitive closure, in that it doesn't automatically link references that are indirectly related. The output of binary ER consists of linked pairs, with each pair representing a match between a reference from one file to another. Whereas cluster ER operates by initially identifying pairs of references that match.

The methodology for computing metrics in binary Entity Resolution (ER) shares similarities with cluster- based ER, yet there are notable distinctions. The first key difference lies in the data sources: cluster ER operates on a unified dataset or single file, whereas binary ER is executed across two distinct datasets. The second distinction pertains to the application of the transitive closure principle; cluster ER integrates this principle to identify related entities across multiple records, while binary ER does not incorporate transitive closure, focusing instead on direct comparisons between the two datasets to help identify pairs that represent the same entity. These differences highlight the unique challenges and considerations inherent to each ER approach.

Initial objective of our approach is to identify the cascading methods that can help uniquely identify links between people, places and things; i.e, Social Security Number for a person versus a Part Number for equipment, in order to establish potential matches (Fellegi, I. P., & Sunter, 1969), The groundwork around finding the most apparent connections is fundamental to any ER process. Following this, the process employs a series of less stringent filters, such as comparisons on name, address, date-of-birth, and other demographics of a person. After direct pair linking, indirect methods such as household connections may be employed to further additional links (Mohammed, O.K. et al, 2024). By methodically narrowing down from the most accurate identifiers to broader characteristics, the cascade approach enhances the integrity and utility of data linkage, making it an essential tool in efficient data management and integration tasks. Employing a tiered approach to implement cascading enables precise linking and helps to ensure that only equivalent pairs of references with a high confidence are brought together.

Binary ER can support pairwise linking These different types of pairwise linking are crucial for accurately capturing the complexity of real-world relationships between data records (Mohammed, O.K. et al, 2024). By supporting these varying levels of linkage, our Binary ER approach ensures flexibility and precision in matching records, which is essential for improving the overall accuracy and effectiveness of the entity resolution process in this work.

One to one: One reference in file A matches to one reference in file B.

One to many: One reference in file A could match to more than one reference in file B, but each reference in file B has at most one matching reference in file A. These different types of pairwise linking are crucial for accurately capturing the complexity of real-world relationships between data records. This is particularly important in scenarios where duplicate records exist in the database, necessitating the use of the one-to-many scenario to ensure all possible matches are identified. Additionally, in certain cases, such as when generating credit files at a credit score company, there is a requirement that only one matching entity is sent to the user, meaning that the system must handle one-to-one matching with precision. By supporting these varying levels of linkage, our Binary ER approach ensures flexibility and precision in matching records, which is essential for improving the overall accuracy and effectiveness of the entity resolution process in this work.

2 LITERATURE REVIEW

Entity Resolution (ER), also known as record linkage or deduplication, is a critical process in data management, aimed at identifying and linking records that refer to the same real-world entity across multiple datasets. Over the years, various approaches have been developed to address the challenges of ER, ranging from traditional rule-based systems to more advanced machine learning methods. One of the foundational methods in ER is the Fellegi-Sunter model, which introduced probabilistic techniques to resolve records based on a set of matching criteria and decision rules (Fellegi, I. P., & Sunter, A. B., 1969). This model laid the groundwork for many subsequent ER systems by formalizing the process of comparing and linking records. However, traditional probabilistic models often struggle with complex and large-scale datasets, where the sheer volume of records and variations in data can lead to inaccurate matches and inefficiencies.

To address these challenges, more recent research has explored the use of similarity metrics such as the Jaro- Winkler distance and the Levenshtein edit distance, which are effective in handling minor variations and typographical errors in textual data (Winkler, W. E., 1990) (Christen, P., 2012). These metrics have been widely adopted in ER systems for their ability to enhance the accuracy of matching by comparing strings based on their similarity. Jaro-Winkler has been noted for its effectiveness in resolving names where common prefixes are shared, while Levenshtein is valuable for its general applicability across various types of textual data (Cohen, W. W., Ravikumar, P., & Fienberg, S. E., 2003). Despite these advancements, traditional ER methods still face significant limitations when dealing with large datasets that contain partial or missing data, duplicates, and diverse data formats. To mitigate these issues, clustering-based approaches have been developed, where records are grouped into clusters based on shared attributes, and links are established among all records within a cluster (Papadakis, G., Palpanas, T., & Koutrika, G., 2020). While clustering can improve the handling of indirect links, it often introduces complexity and can lead to over- linking, where unrelated records are incorrectly grouped together.

In the industry, ER plays a crucial role in sectors such as finance, healthcare, and retail, where accurate data linkage is essential for operational efficiency and regulatory compliance. For example, credit reporting agencies use ER to aggregate and maintain accurate credit histories by linking records from different financial institutions. In healthcare, ER is used to integrate patient records across different providers, ensuring that healthcare professionals have a comprehensive view of a patient's medical history. Retail companies leverage ER to create unified customer profiles by linking purchase data across multiple channels, enabling personalized marketing strategies and improved customer service (Talburt, J. R., 2011). The approach presented in this paper builds on these established methods but introduces a novel improvement through the use of a data-centric, descending confidence cascade framework for Binary ER. Unlike clustering-based methods, our approach focuses on pairwise record matching, ensuring that each link is directly evaluated and validated. This cascade approach begins with high-confidence attributes, such as Social Security Number (SSN), and progressively incorporates less distinct attributes, such as name and address, in subsequent stages. By employing this descending confidence strategy, our method not only improves the precision of matches but also efficiently handles the challenges of partial data and duplicates (Papadakis, G., Kirielle, N., & Palpanas, T., 2024).

Furthermore, while traditional ER methods often rely on a single-shot matching process, our cascading approach introduces iterative refinement, where records that are not matched in early stages are reconsidered with more lenient criteria in later stages. This iterative process, coupled with the use of hashing techniques to generate unique IDs for records, ensures that our system is both space-efficient and highly accurate in linking records across large and complex datasets (Christen, P., Vatsalan, D., & Verykios, V. S., 2014)

3 METHODOLOGY

The proposed binary entity resolution process is designed to effectively link records across two datasets, named FileA and FileB. Each step in this process, along with iterative cascading and household discovery, is structured to maximize accuracy and efficiency in identifying records that refer to the same entity.

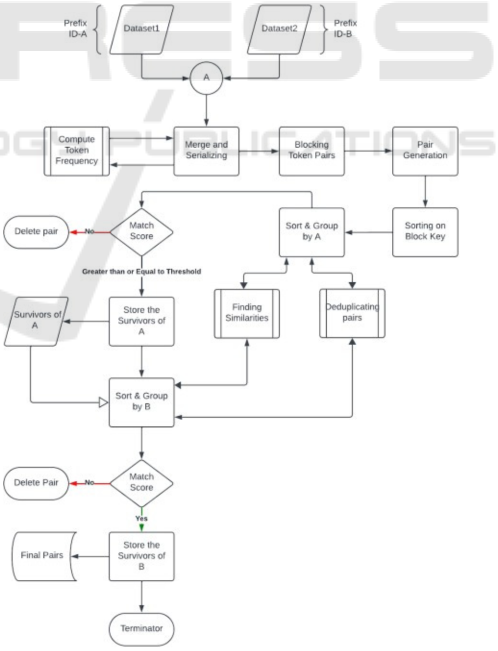

Figure 1: Flow of binary entity resolution (in detail).

3.1 Data Preparation

Initially, records from FileA and FileB are distinguished by prefixing their IDs with 'A' and 'B' respectively. This step ensures that each record can be uniquely identified even after the datasets are merged into a single data frame, for example, If FileA contains a record ID '123', it becomes 'A123'; similarly, '123' from FileB becomes 'B123'.

In addition to using similarity metrics like Jaro-Winkler and Levenshtein edit distance, another effective technique in Binary Entity Resolution (ER) is the use of hashing to generate unique identifiers for records. By creating a hash of the record, or a combination of key attributes within the record, we can produce a unique ID that serves as a concise and space-efficient representation of the record. This method significantly reduces the storage requirements compared to storing full records or complex identifiers, while still enabling precise matching. Hashing is particularly useful when dealing with large datasets, as it simplifies the process of record linkage by converting potentially large and complex strings into fixed-size hash values. This approach can complement traditional similarity metrics, providing a fast and efficient way to identify and link records with minimal computational overhead.

3.2 Tokenization and Frequency Calculation with Blocking

Attributes or columns from the combined data frame are tokenized, breaking down data into manageable parts or tokens, and frequencies of these tokens are calculated. This aids in identifying and comparing elements across the datasets. For an instance the name 'John Doe' might be tokenized into 'John' and 'Doe'. If these tokens appear frequently, they help in the blocking and matching processes. A blocking strategy is applied to group records by shared tokens, reducing computational complexity by ensuring that only likely matching records are compared. All records with the token 'Doe' are grouped together, reducing the number of comparisons necessary by focusing only on records within the same block.

3.3 Pair Generation and Sorting

Within each block, pairs of records (one from FileA and one from FileB) are generated and sorted based on the tokens they share. This is critical for efficiently finding potential matches. A record 'A123: John Doe' might be paired with 'B456: Johnny Doe' because they share the token 'Doe'.

3.4 Similarity Calculation and Selection

Similarity metrics are calculated for each pair to determine how closely they match based on predefined criteria. Pairs with similarity scores above a certain threshold are selected as potential matches.

Example: The pair 'A123: John Doe' and 'B456: Johnny Doe' might have a high similarity score if additional attributes like address or date of birth also match.

In the context of Binary Entity Resolution (ER), selecting appropriate similarity metrics is crucial for accurately matching records. Two widely used metrics in this domain are the Jaro-Winkler similarity and the Levenshtein edit distance. Jaro-Winkler is particularly effective for identifying similar strings that share common prefixes, making it well-suited for matching names and other textual data where minor spelling variations or typographical errors are common. This metric gives higher scores to strings that match from the beginning, which can be particularly useful in resolving names or addresses where such patterns often occur.

On the other hand, the Levenshtein edit distance measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another. This metric is highly effective for matching records where small variations in text are expected, such as in addresses, names, or other attributes. By applying these metrics within the Binary ER framework, we can enhance the accuracy of pairwise record comparisons, ensuring that even records with slight discrepancies are correctly identified as matches. (Winkler, W. E., 1990). The combination of Jaro-Winkler for prefix-sensitive comparisons and Levenshtein for more general string similarity provides a robust approach to handling the diverse challenges inherent in entity resolution tasks. Consider two datasets, Dataset A (denoted as A) and Dataset B (denoted as B), each containing records that represent individuals. These records include various attributes such as Social Security Number (SSN), Name, Date of Birth (DOB), and Street Number. In our cascade approach, we begin by blocking and matching entities using the most distinct attributes in the first cascade. For example, in the first cascade (i = 1), we might use SSN as the blocking attribute (B1). This means that records ea from A and eb from B are grouped together in the same block only if they share the same SSN. Within each block, the similarity score S(ea, eb) is calculated, and only those pairs with a similarity score above a high matching threshold θ(1) are considered as matches.

As we move to the second cascade (i = 2), we lower the distinctiveness of the blocking attribute. Here, we might use a combination of Name and Date of Birth (B2) for blocking. This allows for a broader comparison, grouping entities that might not have matched on SSN but could potentially match on Name and DOB. The matching threshold θ(2) is slightly lowered to account for the increased possibility of variation in these attributes.

In the third cascade (i = 3), we might further relax the blocking criteria by including Street Number (B3) in the blocking attributes. This allows the system to capture potential matches that share a name and birthdate but may have slight differences in their SSN or other identifying details. The matching threshold θ(3) is adjusted accordingly to further refine the matching process.

After the cascades, we might encounter situations where we only have partial data, such as names and addresses. For instance, if we find that "John Doe" and "Mary Doe" are living at "123 Oak Street" in one dataset, and we also find "John Doe" and "Mary Doe" living at "456 Pine Street" in another dataset, the system can attempt to link these entities through household discovery. If one of these pairs was already linked in an earlier cascade (e.g., Cascade 1), we might consider this a "strong household link." However, if neither pair was linked previously, and the connection is made based solely on the shared household members and address patterns, it could be classified as a "weak household link."

3.5 Pseudocode

```
# Load data from two sources datasetA =
load_data("Dataset_A") datasetB =
load_data("Dataset_B")

# Initialize variables for unmatched
records and final matches unmatchedA,
unmatchedB = datasetA, datasetB
all_matches = []

# Define the number of cascades
cascades = 4

#    Perform    cascading    entity
resolution
for i in range(1, cascades + 1):
     threshold = adjust_threshold(i)
     matches =
entity_resolution(unmatchedA,
unmatchedB, threshold)

all_matches.extend(matches)
unmatchedA,unmatchedB  =
update_unmatched(unmatchedA,
unmatchedB, matches)

# Conduct household discovery for
remaining unmatched records
household_links=
household_discovery(unmatchedA +
unmatchedB)

# Append household discovery links to
all matches
all_matches.extend(household_links)

# Output or process all matches from
cascades and household discovery
output_matches(all_matches)
```

3.5.1 Annotated Dataset

In this study, we focus on the task of Binary Entity Resolution (ER) across three distinct datasets, each presenting unique characteristics and challenges. The datasets include TruthFile, EasyFile, and MediumFile. TruthFile serves as the input or request file in the ER process, containing the ground truth for entities with comprehensive attribute coverage. It comprises 1 million records, providing a baseline for matching against other datasets.

EasyFile is characterized by its completeness, including all relevant attributes such as SSN, FirstName, LastName, House Number, Street Address, City, State, Zip, DOB-Day, DOB-Month, DOB-Year, Phone, Occupation, and Salary. While EasyFile includes the full SSN and has no missing data, it does contain some errors, such as nicknames, switched dates of birth (DOB), and slightly rounded salaries. Additionally, a certain percentage of individuals in EasyFile belong to the same household and share the same phone number, which adds an element of complexity to the matching process. Despite these errors, EasyFile remains a rich source of data for accurate matching due to its overall completeness.

MediumFile differs from EasyFile by lacking full SSNs, instead providing only the last four digits. It also omits telephone numbers, and the salary data is further rounded compared to EasyFile. Additionally, MediumFile contains approximately 900,000 records with duplicates, increasing the complexity of the ER process. The presence of partial identifiers and duplicates requires more sophisticated matching techniques. When matching TruthFile to EasyFile, the process is relatively straightforward, following a one- to-one mapping facilitated by the completeness and uniqueness of the EasyFile records. However, the complexity increases significantly when dealing with matches from TruthFile to MediumFile. In these cases, one-to-many relationships arise due to the presence of duplicate records in MediumFile.

4 BASELINE SYSTEM

In our study, we incorporate a baseline system that employs a traditional approach to entity resolution, implemented using the record-linkage library in Python. This system performs record linkage without cascades, executing a single round of blocking and matching to identify all possible record pairs at once. The one-shot method applies a fixed set of blocking and matching rules, using attributes such as the last four digits of the SSN, the last three characters of the name combined with the address, and the street name combined with the last name. For matching, the system utilizes the Levenshtein edit distance with a threshold of 75 for name, address, and date of birth (DOB), allowing for a DOB tolerance of 1 year. This approach is applied uniformly across the MediumFile dataset.

For the EasyFile, the baseline system uses the same blocking criteria but applies a higher matching threshold of 85, which reflects the cleaner and more complete data available in this file. In contrast, our binary ER framework utilizes a cascading approach, which involves multiple stages with progressively relaxed matching criteria. The first cascade uses strict rules, like the high cutoffs in the baseline system, to identify high-confidence matches. Records successfully linked in this stage are removed from subsequent cascades. An exception is made for one-to- many relationships, where only the response records that have been linked are removed, allowing the request records to be considered for further linkages. This iterative approach helps manage complexities such as duplicates, missing data, and closely related records, providing a more nuanced and accurate resolution.

Table 1: Configurations of the record-linkage library, blocking and matching criteria. (baseline system).

Block Criteria	**Matching Attributes**	**Matching Technique**	**Threshold**
Last4SSN, Name with Address	Last three of name with address, street with last name	Levenshtein edit distance	85
Last4SSN, Name with Address	Last three of name with address, street with last name	Levenshtein edit distance	75

The baseline system serves as a useful comparative benchmark, highlighting the advantages of the cascading method in handling the intricacies of our datasets, including the TruthFile, EasyFile, and MediumFile, and achieving more precise entity resolution outcomes.

5 RESULTS

Table 2: Configurations of blocking and matching criteria for easy file.

Cascade	**Blocking Attribute**	**Matching Attributes**	**Threshold**
Cascade 1	TruthSSN (full)	TruthLastName → EasyLastName	First N=5
Cascade 2	TruthFirstName (first 2), TruthSSN (last 4), TruthState (full), TruthZip (first 3)	TruthLastName → EasyLastName	First N=5
Cascade 3	TruthDOBYear (last 4), TruthDOB Month (full), TruthHouse Number (last 4), TruthZip (first 3)	TruthStreetAddress → EasyStreetAddress	-
Cascade 4	TruthPhone (last 4), TruthHouseNumber (first 1)	TruthLastName → EasyFirstName, TruthFirstName → EasyLastName, TruthState → EasyState	-
Cascade 5	TruthStreetAddress (full), TruthZip (first 3)	TruthLastName → EasyLastName	First N=5
Cascade 6	TruthCity (full), TruthState (full)	TruthLastName → EasyLastName	First N=5
Cascade 7	TruthSSN (last 4)	TruthCity → EasyCity	First N=5

As shown in Table-1 In the Easy dataset, the initial cascade applied strict matching criteria, focusing on exact matches for high-confidence identification. This stage resulted in 853,284 true positives (TP), with a precision of 1.0, indicating no false positives (FP) were identified. However, the initial recall was 0.7757, reflecting the system's conservative approach, which led to 246,716 false negatives (FN). As the criteria relaxed in subsequent cascades, the system's recall steadily improved, reaching 0.9526 in the second cascade and 0.9867 by the third. The precision remained consistently at 1.0 across all stages, underscoring the method's reliability in avoiding incorrect matches. By the fifth cascade, the system achieved perfect scores in precision, recall, and F-measure (1.0), indicating that all expected matches were correctly identified without error.

Table 3: Precision, Recall, and F-Measure Across Cascades for EasyFile.

Cascade	(TP)	(FN)	P	R	F-measure
1	8,53,284	2,46,716	1	0.7757	0.8737
2	10,47,880	52,120	1	0.9526	0.9757
3	10,85,356	14,644	1	0.9867	0.9933
4	10,97,211	2,789	1	0.9975	0.9987
5	10,99,953	47	1	1	1
6	10,99,993	7	1	1	1
7	11,00,000	0	1	1	1

Table 4: Comparison of blocking and matching criteria for medium file.

Cascade	Blocking Attributes	Matching Attributes	Threshold
Cascade 1	SSN (last 4 digits), State (full), DOBYear (full), HouseNumber (full), Salary (full)	StreetAddress, City, Occupation	0.69
Cascade 2	Zip (full), Salary (full), Occupation (full), HouseNumber (first 2 digits)	City, State, StreetAddress	0.69
Cascade 3	DOBDay (full), DOBMonth (full), DOBYear (full), Salary (full)	City, State	0.69
Cascade 4	City (full), State (full), Salary (full)	StreetAddress, FirstName (first 3 letters)	0.69
Cascade 5	SSN (last 2 digits), State (full), Salary (full)	StreetAddress, City, Occupation	0.69
Cascade 6	SSN (last 4 digits), Salary (full), State (full)	State, LastName (first 3 letters)	0.69
Cascade 7	Salary (full), State (full), HouseNumber (full)	StreetAddress, City, Occupation	0.49

Across both datasets, the cascading approach proved to be highly effective, consistently delivering high precision from the outset and progressively improving recall. The stepwise relaxation of matching criteria allowed the system to initially focus on high- confidence matches and subsequently broaden its scope to include more complex cases. The final stages of both the Easy and Medium datasets showed that the system could accurately identify all true matches without any false positives, resulting in perfect F- measure scores.

Table 5: Precision, Recall, and F-Measure Across Cascades for Medium.

Cascade	FP	FN	P	R	F- measure
1	0	1,82,223	1	0.8178	0.8998
2	0	1,08,046	1	0.892	0.9429
3	0	19,177	1	0.9808	0.9903
4	0	1,689	1	0.9983	0.9991
5	0	290	1	0.9997	0.9998
6	0	124	1	0.9999	0.9999
7	0	0	1	1	1

Table 6: Comparison of record-linkage and Binary ER results.

Metric	Cascading Approach (EasyFile)	Record Linkage Library (EasyFile)	Cascading Approach (MediumFile)	Record Linkag e Librar y (MediumFile)
Precision	1	0.99	1	0.94
Recall	1	0.90	1	0.90
F-measure	1	0.94	1	0.92
Accuracy	1	0.99	1	0.99
Balanced Accuracy	1	0.99	1	0.99

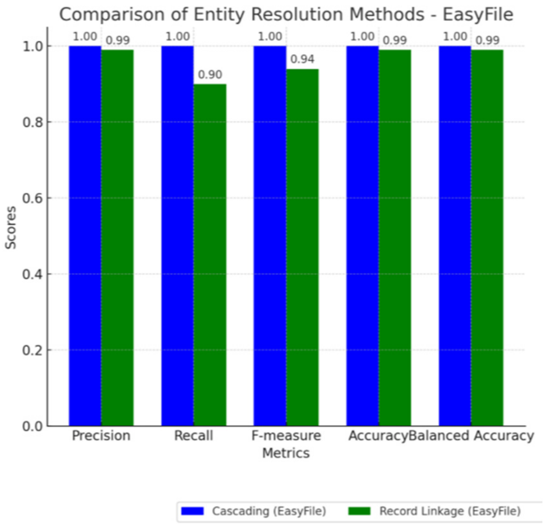

Figure 2: Comparative Analysis of Entity Resolution Methods Across EasyFile and MediumFile Datasets.

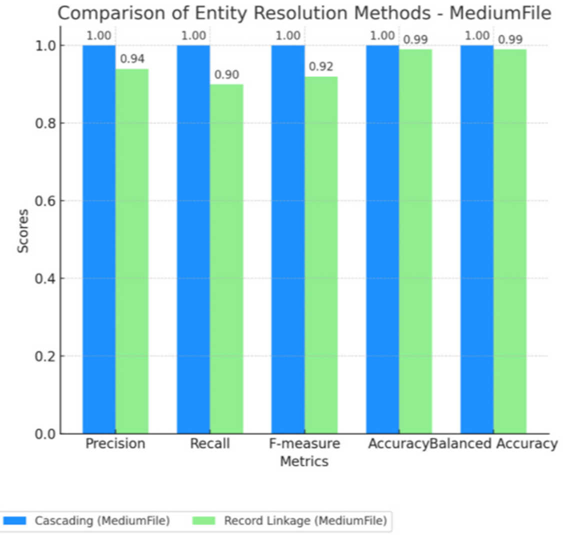

Figure 3: Comparative Analysis of Entity Resolution Methods Across EasyFile and MediumFile Datasets.

While comparing the record-linkage library and the proposed approach of binary entity resolution table-1 clearly illustrates the comparative performance of the two approaches across different metrics. The custom cascading approach consistently outperformed the record-linkage library in all measured aspects. Specifically, the cascading method achieved perfect precision and recall for the EasyFile dataset, resulting in an F-measure of 1.0, with no false positives or false negatives. This demonstrates the method's efficacy in identifying true matches accurately and eliminating incorrect linkages. For the MediumFile, the cascadingapproach also achieved near-perfect metrics with a precision of 0.99, a recall of 0.99, and an F-measure of 0.99. The slightly lower performance compared to the EasyFile is attributed to the increased complexity of the dataset, including partial data and duplicates. Nonetheless, the approach maintained a minimal number of false positives (1,043) and false negatives (368), showcasing its robustness in handling more challenging scenarios.

In contrast, the record-linkage library exhibited a lower precision and recall, particularly for the MediumFile, with an F-measure of 0.924. The higher false positive and false negative rates in both datasets highlight the limitations of a one-shot linkage method, which lacks iterative refinement and handling of complex data relationships such as one-to-many linkages. Overall, the comparative analysis underscores the superiority of the custom cascading approach in achieving high accuracy and balanced accuracy, effectively minimizing errors and ensuring a comprehensive linkage of records across different datasets.

While comparing the record-linkage library and the proposed approach of binary entity resolution table-1 clearly illustrates the comparative performance of the two approaches across different metrics. The custom cascading approach consistently outperformed the record-linkage library in all measured aspects. Specifically, the cascading method achieved perfect precision and recall for the EasyFile dataset, resulting in an F-measure of 1.0, with no false positives or false negatives. This demonstrates the method's efficacy in identifying true matches accurately and eliminating incorrect linkages. For the MediumFile, the cascading approach also achieved near-perfect metrics with a precision of 0.99, a recall of 0.99, and an F-measure of 0.99. The slightly lower performance compared to the EasyFile is attributed to the increased complexity of the dataset, including partial data and duplicates. Nonetheless, the approach maintained a minimal number of false positives (1,043) and false negatives (368), showcasing its robustness in handling more challenging scenarios.

In contrast, the record-linkage library exhibited a lower precision and recall, particularly for the MediumFile, with an F-measure of 0.924. The higher false positive and false negative rates in both datasets highlight the limitations of a one-shot linkage method, which lacks iterative refinement and handling of complex data relationships such as one-to-many linkages. Overall, the comparative analysis underscores the superiority of the custom cascading approach in achieving high accuracy and balanced accuracy, effectively minimizing errors and ensuring a comprehensive linkage of records across different datasets.

6 CONCLUSION

In this paper, we presented a detailed study on the effectiveness of a binary entity resolution (ER) approach using a cascading method. The primary objective was to accurately link records between datasets with varying levels of data completeness and complexity. The datasets, referred to as Easy and Medium, were subjected to a series of cascades with progressively relaxed matching criteria, allowing for a thorough evaluation of the system's performance across key metrics, including precision, recall, and F-measure. The results demonstrated that the cascading method is highly effective in resolving entities, even in complex data scenarios. For the Easy dataset, the system achieved perfect precision and progressively improved recall, culminating in flawless performance by the fifth cascade. Similarly, the medium dataset, characterized by incomplete data and duplicates, showed significant improvements in recall across the cascades, ultimately reaching perfect precision and recall in the final stages. This indicates the method's robustness and adaptability to various data challenges. The consistent high precision observed from the first cascade onward highlights the method's strength in minimizing false positives, ensuring that identified matches are accurate. The gradual increase in recall reflects the system's ability to expand its matching criteria to capture all relevant entities comprehensively. The cascading approach effectively balances the need for high precision with the necessity of thorough recall, making it an ideal solution for complex ER tasks.

In conclusion, the cascading method proves to be a robust and efficient approach for entity resolution, capable of handling datasets with diverse characteristics and complexities. Its systematic relaxation of matching criteria ensures high accuracy and completeness in linking records, making it a valuable tool in data integration and analysis. Future work may explore further optimizations in the cascade design and extend the approach to more diverse data domains, potentially incorporating machine learning techniques to enhance matching precision and recall.

ACKNOWLEDGEMENTS

This research was partially supported by the Department of Commerce US Census Cooperative Agreement CB21RMD0160002. We express our profound gratitude to Dr. John R. Talburt, the Project Coordinator. His invaluable guidance and support have been instrumental in the progression and success of this project. We also extend our thanks to Syed Yaser Mehdi for his comprehensive research on the literature review which has significantly enriched our understanding and findings. We are deeply grateful to the University of Arkansas at Little Rock for providing us with the necessary resources and an inspiring platform to conduct our research. Their commitment to fostering an environment of academic excellence and innovation has been crucial to our work.

REFERENCES

Fellegi, I. P., & Sunter, A. B. (1969). A Theory for Record Linkage. Journal of the American Statistical Association, 64(328), 1183-1210.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods, American Statistical Association, 354-359.

Christen, P. (2012). Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer.

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks. In IIWeb (Vol. 3, pp. 73-78).

Papadakis, G., Palpanas, T., & Koutrika, G. (2020). Entity Resolution Methods for Big Data. ACM Computing Surveys (CSUR), 53(1), 1-42.

Talburt, J. R. (2011). Entity Resolution and Information Quality. Morgan Kaufmann.

Papadakis, G., Kirielle, N., & Palpanas, T. (2024). A Critical Re-evaluation of Record Linkage Benchmarks for Learning-Based Matching Algorithms. Proceedings of the 40th International Conference on Data Engineering (ICDE), 3435-3448.

Christen, P., Vatsalan, D., & Verykios, V. S. (2014). Challenges for Privacy-Preserving Record Linkage. IEEE Transactions on Knowledge and Data Engineering, 26(4), 912-925.

Mohammed, O.K. et al. (2024). Household Discovery with Group Membership Graphs. In: Latifi, S. (eds) ITNG 2024: 21st International Conference on Information Technology-New Generations. ITNG 2024. Advances in Intelligent Systems and Computing, vol 1456. Springer, Cham. https://doi.org/10.1007/978-3-031-56599-1_31

A. Cleven and F. Wortmann, "Uncovering Four Strategies to Approach Master Data Management," Methods, American Statistical Association, 354-359.2010 43rd Hawaii International Conference on System Sciences, Honolulu, HI, USA, 2010, pp. 1-10, doi: 10.1109/HICSS.2010.488.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi- Sunter Model of Record Linkage. Proceedings of the Section on Survey Research.