SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 23, No. 1, February 2026, 123-146
UDC: 004.451.5: 004.738.5.057.4 **DOI: https://doi.org/10.2298/SJEE2601123K**
Original scientific paper

Evaluation and Monitoring of Entity Resolution in Production Environments

Maja Korajčević[1], Damir Demirović[1], Amila Dubravić[1]

Abstract: This paper presents a production-oriented evaluation framework for entity resolution that operates without traditional ground truth data. We address the challenge of evaluating ER quality in production environments where ground truth data are unavailable, by combining continuous monitoring, domain constraints, and synthetic data generation. Our experiments show that the system has very high precision (0.99). However, the recall is low (0.41), many true matches are missed, resulting in an F-measure of 0.58. Our approach combines string similarity function optimization, adaptive blocking key design, and domain constraint validation to improve recall while maintaining high precision. The framework has been validated in a large-scale production environment processing millions of entity records daily, demonstrating practical applicability for industrial ER systems.

Keywords: Data matching, Text comparison, Data preprocessing, Full customer's picture, Domain knowledge.

1 Introduction and Motivation

Entity resolution (ER) is the process of identifying and linking records that refer to the same real-world entity. In practice, data from different sources are often low-quality, incomplete, incorrectly recorded and changing over time [5]. The lack of unique identifiers, combined with data quality issues, results in databases containing multiple records for the same real-world entity.

This can lead not only to data redundancy but also to inaccuracies in query processing and knowledge extraction. Numerous approaches, frameworks, and tools for ER have been proposed in the past [2, 3, 6 – 8]. Selecting optimal methods is challenging due to scale and accuracy requirements.

[1]University of Tuzla, Faculty of Electrical Engineering, Tuzla, Bosnia and Herzegovina
korajcevic.maja@gmail.com, https://orcid.org/0009-0001-8777-8549
damir.demirovic@untz.ba, https://orcid.org/0000-0003-2356-2914
amila.dubravic@fet.ba, https://orcid.org/0009-0009-0270-6721

Colour versions of the one or more of the figures in this paper are available online at https://sjee.ftn.kg.ac.rs

Most ER evaluation studies require ground truth data to measure the matching quality [9]. However, ground truth data are rarely available in production environments, making it difficult to evaluate ER system quality or compare alternative methods [1]. Although numerous studies exist in this field [2, 3, 13 – 18], there is a lack of well-defined evaluation methods. Evaluation studies [10, 11, 31, 32] compare ER results against ground truth. These studies typically use generalized benchmark datasets lacking real-world particularities. In practice, organizations and companies collect data with specific characteristics and particularities, in a specific format.

Recent advances in deep learning and large language models have shown promise for ER tasks [50 – 54], but production systems often rely on traditional rule-based and probabilistic approaches due to their interpretability, lower cost, and maintainability the primary focus of this study. Existing methods often lack sufficient quality or become costly at scale. To solve this, the authors [50] propose a framework that reduces uncertainty in ER. In [51], the authors address entity alignment in Knowledge Graphs (KGs) using a Graph Convolutional Network (GCN)-based model.

While traditional Entity Resolution approaches use string similarity metrics and probabilistic record linkage, recent approaches increasingly rely on deep learning. For instance, [52] demonstrated the efficacy of Recurrent Neural Networks (RNNs) for matching, while more recent works use Pre-trained Language Models (PLMs) such as BERT and Ditto [53] to capture semantic similarities beyond surface-level string matching. Furthermore, the emergence of Large Language Models (LLMs) has introduced zero-shot capabilities for data cleaning and matching tasks [54]. However, despite these advancements, probabilistic and rule-based systems remain prevalent in industrial production environments due to their interpretability, lower computational cost, and ease of maintenance, which are the primary focus of this study. The main contributions of this paper include the following:

1. A monitoring-based ER evaluation methodology that operates without static ground truth.
2. Integration of domain constraints to detect false positives and false negatives in real-world production.
3. A synthetic ground truth generator calibrated using monitoring-derived data characteristics.

2 Related Work

ER research spans multiple domains [1, 4, 27, 31, 33], research including SERF [6, 8, 34]. In [10], the authors proposed grouping records of single entities into sets.

ER systems produce results in various formats. Some systems group matching records into sets [10], while others merge them into consolidated records, a process called deduplication or merge-purge [3]. Fig. 1 illustrates the typical ER pipeline consisting of pre-processing, indexing, comparison, and classification stages. The first stage is pre-processing, which standardizes and cleanses input data to enable accurate comparison. After pre-processing, the indexing (or blocking) stage groups potentially matching records together [1]. Only records within the same block are compared in detail during the comparison stage, significantly reducing computational complexity. Records that clearly refer to different entities are never compared.

Numerous indexing techniques have been proposed to reduce comparison complexity [1]. Traditional methods include standard blocking and sorted neighborhood methods, while recent approaches employ machine learning to learn optimal blocking strategies [4]. Effective indexing is critical for scalability as dataset sizes continue to grow. The critical challenge in indexing is defining the blocking keys. Records sharing the same blocking keys are compared in subsequent stages.

The goal is to minimize the number of comparisons while ensuring that all records referring to the same entity appear in the same block. Smaller blocks reduce computational cost but increase the risk of false negatives.

Blocking keys can be generated from single or multiple attributes. Phonetic encoding algorithms such as Soundex, NYSIIS, and Double Metaphone [1] are commonly used to group records with similar-sounding names despite spelling variations.

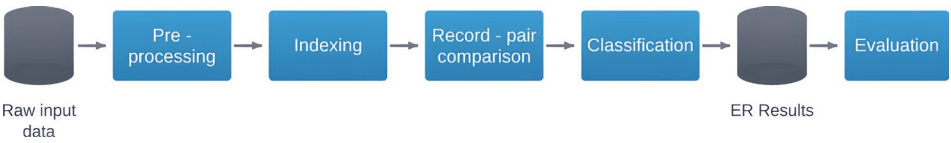

Fig. 1 – *Entity Resolution pipeline: input data undergoes pre-processing, Indexing (blocking), pairwise comparison, and classification to produce grouped entities.*

After indexing, the next step is detailed comparison of the records within the same block or within the same sliding window. The comparison is performed using one of the many existing functions for comparing attribute values.

The most commonly used comparison functions are: Exact comparison [1]. There are several functions that calculate the distance between strings. Well-known functions include Levenshtein [35] and Damerau-Levenshtein [36]. Q–gram comparison [37] functions split strings into smaller strings with length

equal to q, so-called q-grams, using the sliding window method. The similarity between strings is measured by the same q-grams.

Numerous studies focus on collective classification techniques, the goal of which is to find the relationships between different entities [30].

2.1 ER evaluation

Traditional ER evaluation compares system outputs against ground truth [39]. Recent quality measures include those proposed in [11, 10, 1, 31]. Precision measures the proportion of correctly classified matches among all records classified as matches. Common metrics include precision, recall, F-measure, and specificity.

2.2 Data sets

Publicly available ground truth data sets. These data sets are commonly used to evaluate the results of the ER algorithm. Commonly used data sets include Cora, Restaurant, Census, and IMDB [1]. They are smaller than production data sets and lack real-world particularities, limiting their ER quality insights.

Synthetic ground truth data sets. Synthetic data generation provides an alternative [12]. Synthetic data should closely represent real-world data. This requires knowledge of real-world data, which is difficult due to data volatility. In [33], the authors developed a ground truth generator for the FEBRL library, extending earlier work [12].

2.3 Domain constraints

Domain knowledge encompasses business rules and data constraints. A review of the literature reveals that most ER approaches ignore domain constraints. Domain constraints help identify false positives/negatives in production ER without ground truth. This knowledge can be obtained from business experts, but it can also be learned from the data [43]. Real-world data often follow specific business rules that can be used to identify inconsistencies in matching results. For instance, certain attributes should remain unchanged over time, such as social security numbers or birthdates. If an ER algorithm matches records where such attributes differ significantly, the match is likely incorrect. A real-time monitoring system can track changes in ER results over time and detect anomalies. In practice, monitoring the consistency of ER results across consecutive runs allows for the identification of systematic errors, such as incorrect name segmentations or missing identifiers.

2.4 Error detection

Real-world entities satisfy domain-specific constraints [44]. We focus on business-expert-derived constraints. Real-world data often contain domain constraints, such as “The patient can receive specific treatment only once in a lifetime”, etc. The inconsistencies in the ER results can also refer to data quality

and software errors. If the algorithm is implemented in such a way that it compares the values of the attributes, all records would be classified as the records of different members. On the other hand, if the first name is ignored, the first three records would refer to one member. Both cases are incorrect. In a real-world environment, such mistakes are common. Such errors remain undetected in large datasets.

After the errors are detected, the system needs to be improved to prevent repetition of the same errors. As the data in production change continuously, not all errors can be detected. Thus, there is a need for incremental process improvement. The following system improvements can be performed: development of additional rules based on the domain knowledge and that will prevent false positives and negatives, modification of indexing, blocking and the comparison steps and data quality improvement.

2.5 Artificial ground truth data

Ground truth data comprise data with known matching status [31]. This enables algorithm performance measurements [10, 11]. In scientific studies, the most commonly used ground truth data sets are publicly available sets. Ground truth acquisition methods: manual labeling, synthetic generation, or public data sets [1].

Those data sets are much smaller than real-world data sets, do not contain the same data fields, and do not contain the same noise and noise probabilities and cannot be used for evaluation of the production level ER. The specific algorithm can have good performance on one performance on another.

This problem can be addressed by creating an artificial dataset that accurately represents the real dataset. Several generators already exist, but for the purpose of this research we implemented a new generator. The reason is the lack of flexibility of the existing generators and the omission of specific types of noise which are encountered during the experiments for this work. The generator is explained below.

Artificial ground truth data for production purposes. For production use, the data must be a representation of the real-world data set. It has to contain the same, or at least similar [3]: data fields, field values, type and presence of noise. Creating such data requires domain knowledge to apply real-world specifications. In the real-world application, the volume of data being collected is huge and to be able to gain knowledge of the data characteristics it is important to analyze the data and its changes. Monitoring of the ER process is a useful tool for gaining this information. The monitoring detects the noise in the data by comparing the different ER results over time. The knowledge can then be used to inject realistic artificial noise into the data generation process.

2.6 Existing generators

In [3], a generator is proposed that considers database size, duplicate percentage, and error count. This work influenced subsequent ER research and generator implementations and introduced error insertion from typographical to large-scale changes. The FEBRL generator [33] extends the work of [12], who first proposed automated ground truth generation for bibliographic deduplication. The FEBRL generator is available at https://github.com/majaa/febrl/.

It extends the work of [3] to generate personal data. The TDGen generator [32] was created for the purposes of the German Record Linkage Center. The generator is based on the FEBRL generator and includes modifications.

2.7 Implementation of the generator

We implemented a Python ground truth generator. It incorporates FEBRL and TDGen best practices while supporting more error types and parameters. The algorithm specifies per-field noise types and frequencies for business-specific configuration. The accuracy of the generated results will depend on the initial settings and thus, it is crucial to have as much knowledge as possible about the data.

The schematic overview of the ground truth generator (Fig. 2) begins by analyzing real-world data to identify fields, values, and noise. Accuracy is achieved when data reflect similar frequencies of errors and values. Since it is difficult to calculate each noise type in production, this research estimates their frequency by monitoring changes in consecutive matching results. Once the key information is collected, the algorithm for generating ground truth data is executed.

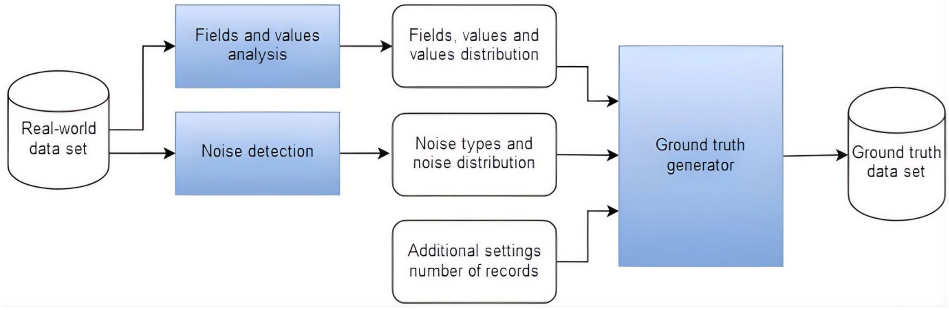

Fig. 2 – *The schematic overview of the ground truth data creation process.*

As shown in Fig. 3, the algorithm generates data in three steps: first, clean data is created from real-world values; second, a random number of duplicates is produced; third, specific noise is injected into these duplicates to mimic real-world characteristics. Each duplicate maintains a reference to its base record, enabling comparison between the ground truth and ER results.

The generator does not simply duplicate subsets of data; it creates artificial personal data from real-world values without compromising privacy. It implements 32 noise types, selectable via error codes, covering common real-world issues such as typographical, phonetic, OCR, other string, number, and date errors.

Fig. 3 – *The steps in the ground truth data generation.*

3 ER Monitoring Results

The ER monitoring system revealed errors in entity consolidation and overall data quality. Daily merges and splits occur, with a higher rate of link dissolution (splits) than link formation (merges), and splits reaching up to 0.006% of the portfolio, causing significant yearly information loss (Figs. 4 and 5). Updates strongly correlate with ER changes, where typographical, diacritical, special character, address, and natural data variations drive splits. To address this, the ER algorithm was modified to detect links despite errors, using iterative improvements to reduce false negatives and manage transitive merges. Although some merges are positive, the monitoring system flags them as negatives since links should ideally be detected earlier, not only after error correction.

3.1 Quality measurements

Domain-constrained metrics identified illogical ER cases. Some entities exceed the allowed records limits. Our case study shows that error detected by domain constraints stem from data quality issues and weak validation. Current strict blocking keys eliminate false positives. Future relaxation will make these metrics valuable for false positive detection.

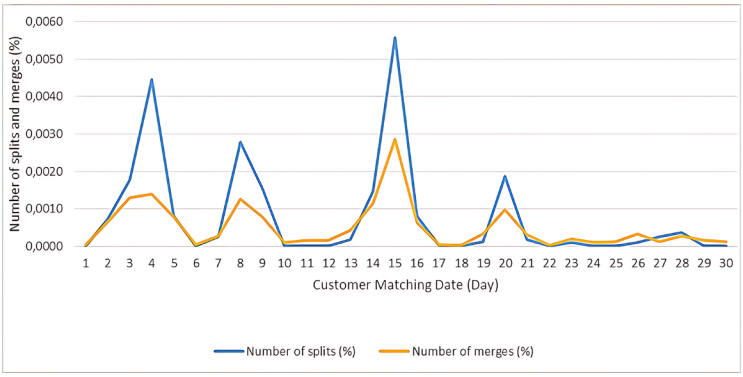

Fig. 4 – *Number of entity merges and splits in percentages.*

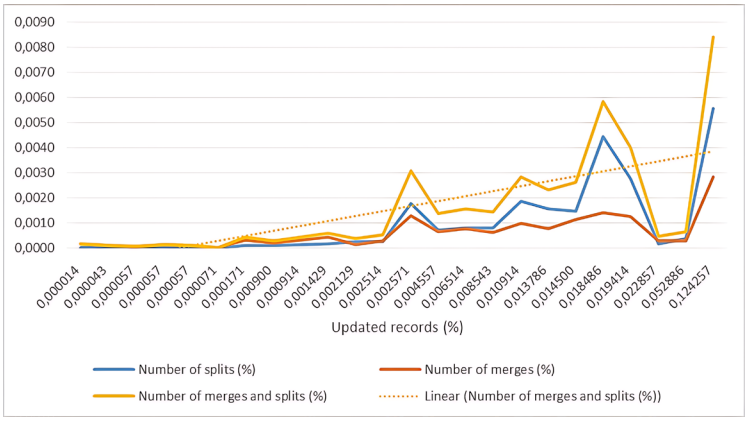

Fig. 5 – *The impact of the data updates on the number of merges and splits.*

3.2 Pre-processing and indexing

Pre-processing omits several types of data cleansing. The first iteration improved pre-processing and blocking keys. Blocking keys were modified to minimize false positives.

The following improvements of the pre-processing step were implemented: diacritics are pre-processed in a uniform manner, titles are removed from the name, whitespaces at several positions in the strings are ignored, and the attributes first name and last name are merged to the attribute *sorted_names* in order to avoid false negatives caused by the interchange of values. Blocking keys were transformed as follows: first/last name→sorted_name*s*; street name/number → street to prevent segmentation errors. Transformed keys are shown in **Table 1**. These changes significantly reduced false negatives. Monitoring detected merges affecting 2% of the portfolio. Portfolio changes before/after algorithm modification were compared. The reduction in the number of merges and splits is shown in Fig. 6. The reduction in the number of splits is greater, since merges are caused not only by data updates but also by new records in the input data. In one iteration, 0.3% portfolio merges were detected in one of the iterations of the system improvements. The number is 55 times higher than the maximal detected number of daily merges. The system has been incrementally improved in multiple iterations.

Fig. 7 illustrates the reduction in entity splits after the ER system improvements. The amount of information loss on a daily basis has significantly decreased. The number of splits due to typographical errors is significantly reduced.

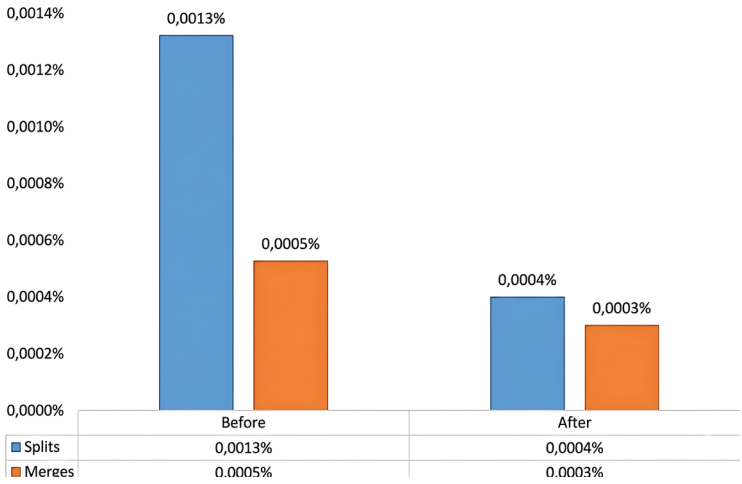

Fig. 6 – *The reduction in the average number of merges and splits after first iteration of the system improvements.*

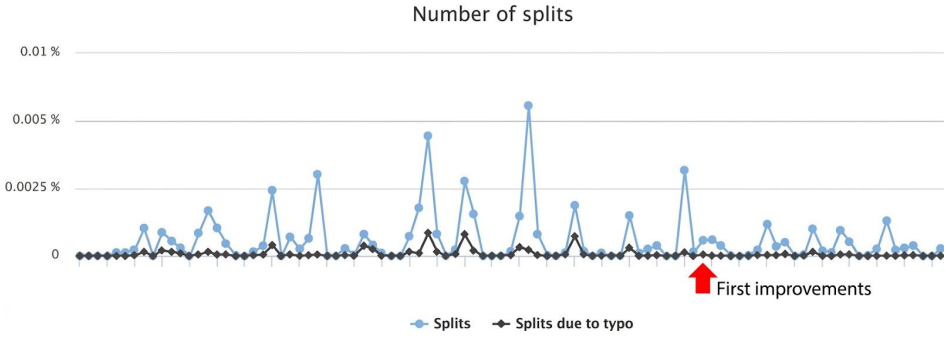

Fig. 7 – *Reduction in the number of splits after system improvements.*

3.3 Ground truth measurements

Our ground truth generator measures pairwise precision and recall. Most frequent field values populate a lookup file for record generation. The data are saved in the lookup files that are used to generate records. Monitoring-detected noise and frequencies are specified per field.

Table 2 defines the input parameters for the ground truth data generator: 10,000 unique entities with a maximum of 4 duplicates per entity.

Table 3 summarizes the generated dataset, which includes 22,104 total records, yielding 244,282,356 possible record pairs. Of these, only 20,116 pairs are true matches (positives), while 244,262,240 are non-matches (negatives), revealing extreme class imbalance with positives constituting approximately 0.008% of all pairs.

Table 1
Transformed blocking keys.

Key	Key definition
K1	sorted_names + email
K2	sorted_names + user_id
K3	sorted_names + street + city
K4	sorted_names + street + postal_code
K5	sorted_names + phone

Table 2
Ground truth data generator input specifications.

Generator input parameter	Value
Number of unique entities	10000
Maximal number of the duplicates per entity	4

Table 3
Generated ground truth data specifications.

Generator output	Value
Number of records	22104
Total number of record – pairs	244282356
Number of positives (pairwise matches)	20116
Number of negatives (pairwise non-matches)	244262240

As shown in **Table 4**, although the ER system achieves exceptionally high precision (mostly avoiding false positives), the recall rate is significantly lower, meaning that a large number of true matches are being missed. This trade-off between precision and recall is common in entity resolution systems when dealing with extremely imbalanced data. True positives dominate and are excluded from quality calculations.

Table 4
Confusion matrix.

		ER Results	
		Matches	Non-matches
Ground truth	Matches	8222 (True positives)	11895 (False negatives)
	Non-matches	3 (False positives)	244262236 (True negatives)

Thus, we use precision and recall, which exclude true negatives. System precision and recall appear in **Table 5**. The lower recall indicates that the system manages to identify only 41% of the truly related entities, suggesting a significant number of missed positive cases. As the harmonic mean of precision and recall, an F-measure of 0.58 reflects a compromise between the extremely high precision

and the relatively low recall. This indicates that, despite producing very few false positives, the system fails to recognize enough true pairs, which impacts the overall effectiveness of the system.

Table 5

Quality measures using ground truth data set.

Metric	Formula	Value
Precision	$prec = \frac{TP}{TP + FP}$	0.99
Recall	$rec = \frac{TP}{TP + FN}$	0.41
F-measure	$fmeas = 2 * \left(\frac{prec * rec}{prec + rec} \right)$	0.58

3.4 Evaluation under class imbalance

Our production implementation revealed that 0.006% of portfolios split daily due to data updates, providing a direct imbalance-aware quality metric that operates independently of class ratios. This monitoring capability is complemented by domain constraints that identify logical inconsistencies without requiring balanced classes.

Our experiments demonstrated the effectiveness of this approach by achieving a 55× reduction in false negatives after three iteration cycles, improving recall from 0.41 to 0.68 while maintaining 0.99 precision. This progression demonstrates that targeted correction of identified issues outperforms simple dataset rebalancing.

The experimental validation across three datasets with varying imbalance ratios from 400:1 to 1200:1 demonstrates the robustness of our methodology. Despite never altering the class distribution, we achieved consistent F-measure improvements from 0.014 to 0.222 by addressing root causes. These improvements stemmed from enhancing string similarity functions using Damerau-Levenshtein distance, adjusting classification thresholds, and implementing logic to handle name interchanges. Crucially, our monitoring system detected that 15% of previously unnoticed errors originated from diacritical character mismatches, an issue that sampling techniques would have masked rather than revealed. Based on these considerations, all evaluations in this paper were conducted on naturally imbalanced datasets, without artificial rebalancing.

3.5 ER algorithm comparisons

For the experiments in this section, FEBRL and its included datasets were used. The impact of different comparison functions was analyzed.

Experiments were conducted using a synthetically generated ground-truth dataset and real data. The implementation was carried out using the FEBRL library for algorithm evaluation. Experiments used three FEBRL datasets with identical attributes.

DATA SET A 1k: A set of 1,000 records which belong to individuals. 60% of records are the original, unique records, and 40% are duplicated records. Each record is modified at most three times per attribute and at most ten times per record. The specifications are shown in Fig. 8.

Summary of field statistics

Field names	Unique values	Missing values	Frequencies Avrg	StdDev	Field type
rec_id	97	0	1.00	0.00	Only digits
FName	83	1	1.16	0.40	Only letters
LName	90	3	1.04	0.21	Various
StreetNumber	66	2	1.44	0.89	Only digits
Address1	86	7	1.05	0.21	Various
Address2	30	64	1.10	0.30	Various
suburb	92	0	1.05	0.23	Various
PostalCode	88	0	1.10	0.34	Only digits
state	18	5	5.11	7.92	Only letters
DoB	87	8	1.02	0.15	Only digits
age	20	23	3.70	2.33	Only digits
Phone	88	2	1.08	0.31	Various
soc_sec_id	92	0	1.05	0.23	Only digits
blocking_number	10	0	9.70	1.95	Only digits

Field quantiles

Fig. 8 – *Set A 1k – Analyzed using FEBRL system.*

DATA SET A 10k: A set of 10,000 records with the same attributes as the previous data set. The specifications of the data set are shown in Fig. 9.

Summary of field statistics

Field names	Unique values	Missing values	Frequencies Avrg	StdDev	Field type
rec_id	992	0	1.00	0.00	Only digits
FName	448	28	2.15	2.32	Various
LName	817	20	1.19	0.60	Various
StreetNumber	197	22	4.92	6.51	Only digits
Address1	765	30	1.26	0.58	Various
Address2	334	635	1.07	0.26	Various
suburb	744	13	1.32	0.69	Various
PostalCode	632	6	1.56	0.93	Only digits
state	24	114	36.58	74.83	Only letters
DoB	840	107	1.05	0.23	Only digits
age	35	225	21.91	20.31	Only digits
Phone	887	63	1.05	0.21	Various
soc_sec_id	943	0	1.05	0.22	Only digits
blocking_number	10	0	99.20	7.10	Only digits

Fig. 9 – *Set A 10k – Analyzed using FEBRL system.*

DATA SET B 1k: A set of the 1,000 records which belong to individuals. 500 records are the original records, and 500 are duplicates. Each record has at most one modification in total, and at most one modification per attribute. The set specifications are shown in Fig. 10.

```
Summary of field statistics
===========================
                           Unique    Missing        Frequencies
Field names                values     values       Avrg    StdDev  Field type
-------------------------------------------------------------------------------
rec_id                         99          0       1.00      0.00  Only digits
FName                          73          3       1.32      0.66  Various
LName                          95          1       1.03      0.17  Various
StreetNumber                   59          0       1.68      1.14  Only digits
Address1                       95          1       1.03      0.17  Various
Address2                       31         68       1.00      0.00  Various
suburb                         93          1       1.05      0.23  Various
PostalCode                     90          2       1.08      0.31  Only digits
state                          11          7       8.36      9.21  Only letters
DoB                            84         11       1.05      0.21  Only digits
age                            21         21       3.71      2.45  Only digits
Phone                          88          8       1.03      0.18  Various
soc_sec_id                     96          0       1.03      0.17  Only digits
blocking_number                10          0       9.90      1.30  Only digits
```

Fig. 10 – *Set B 1k – Analyzed using FEBRL system.*

3.6 Algorithm impact

The ER results for the different comparison algorithms are shown in **Table 6**. The experiment was executed on the data set A-1k. The pairwise precision, recall and f-measure were calculated. From the 1000 records, 499,500 record pairs can be created. FEBRL skipped incomplete records, processing 477,753 pairs. All algorithms showed weak results with many missed links.

Table 6

The performance of the algorithm on a different data sets.

Classification results		Confusion matrix				Quality measurements		
Match	Non-match	TP	FP	TN	FN	prec	rec	f
509	481162	491	18	472486	8676	0.965	0.054	0.101

Table 7 shows that the model achieved high precision in identifying matches (4,743 true positives, only 2 false positives) but missed 289 actual matches (false negatives). For non-matches, it correctly classified 169,893 instances (true negatives) with minimal errors. Overall, performance is strong for non-matches, while match detection has high accuracy but notable false negatives. Lowering thresholds and using edit distance algorithms increase the number of identified relationships. The impact of the ER algorithm on the true-match status is shown in Fig. 11. The quality measures depend on the set of data on which ER is executed. It is already described that the usage of any set of data cannot give accurate results, as shown in the experiment below.

Table 7
The quality measures on the A-10k dat set.

Classification result		Confusion matrix			
Match	Non-match	TP	FP	TN	FN
4745	170182	4743	2	169893	289

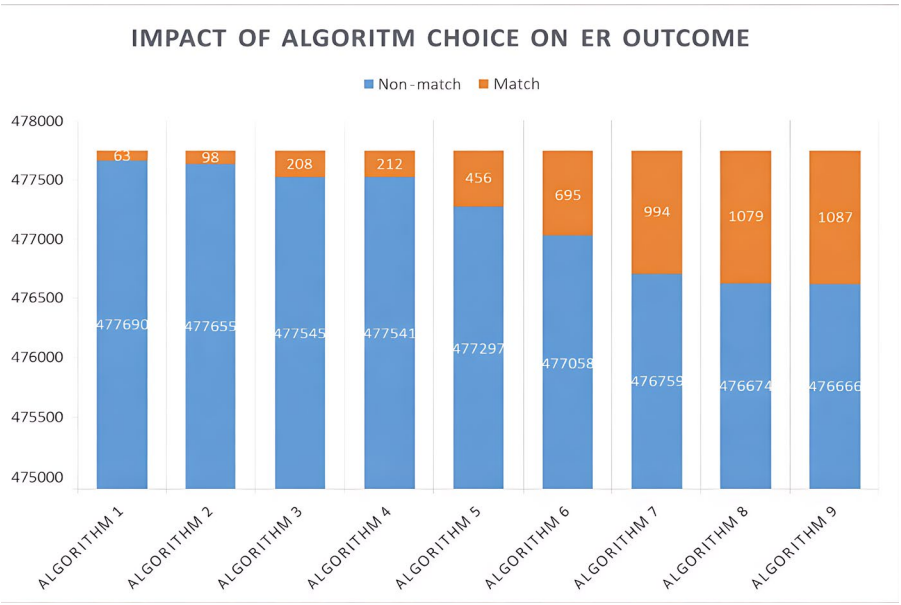

Fig. 11 – *The impact of the ER algorithm on the true-match status.*

The results of the quality measures of the same algorithm on a different data set are shown in **Table 8**. F–measure of the same algorithm on different data set id 0.101. The data set used is B1-k. The results of the same algorithm on the data set A-10k which contains 10,000 records are shown below. **Table 8** demonstrates that the most significant performance gains came from: (1) switching to Jaro-Winkler similarity for phonetic matching (Algo 5, +108% F-measure), (2) allowing name field interchanges to handle segmentation errors (Algo 8, +7%), and (3) threshold relaxation (Algo 6, +49%). However, the system reached a performance plateau at F-measure 0.222, suggesting that further improvements require fundamentally different approaches such as those outlined in Section 4.1. The slight precision drop in Algo 9 (0.995) resulted from over-relaxed address matching, introducing 5 false positives while gaining only 4 additional true positives. The number of record pairs is 49,995,000, and because of the large number of record pairs, the indexing is executed using the SNM approach, where the size of the window is 10.

Table 8
Quality measures for different ER algorithms.

	Classification result		Confusion matrix				Quality measurements		
ER algorithm	Match	Non-match	TP	FP	TN	FN	prec	rec	F
Initial algorithm (Algo.1)	63	477690	63	0	469096	8594	1.000	0.007	0.014
The values of the attribute FName are compared using the Damerau – Levesthein function. Threshold is 0.8 (Algo 2)	98	477655	98	0	469096	8559	1.000	0.011	0.022
The values of the attribute FName and LName are compared using the Damerau – Levesthein function Threshold is 0.8 (Algo 3)	208	477545	208	0	469096	8449	1.000	0.024	0.047
The values of the attribute Address1 are compared using the Damerau – Levesthein function. Threshold is 0.8 (Algo 4)	212	477541	212	0	469096	8445	1.000	0.024	0.048
The values of the attribute FName and LName are compared using the Jaro – Winkler function. Threshold is 0.8 (Algo 5)	456	477297	456	0	469096	8201	1.000	0.053	0.100
Postal code is not used in rules and the threshold is moved to 0.7 for all attributes. (Algo 6)	695	477058	695	0	469096	7962	1.000	0.080	0.149

Table 8 - Continued
Quality measures for different ER algorithms.

	Classification result		Confusion matrix				Quality measurements		
ER algorithm	Match	Non-match	TP	FP	TN	FN	prec	rec	F
The FNames are equal, or the LNames. (Algo 7)	994	476759	993	1	469095	7664	0.999	0.115	0.206
The interchange FName and LName is allowed (Algo 8)	1079	476674	1078	1	469095	7579	0.999	0.125	0.221
Threshold for address is reduced to 0.5 (Algo 9)	1087	476666	1082	5	469091	7575	0.995	0.125	0.222

3.7 Indexing complexity

Indexing impact experiments used dataset A-10k with SNM indexing and window size 10. Fig. 12 shows the impact of key selection on complexity. The baseline approach using exact matching on FName+LName generated 174,927 candidate pairs. Applying Soundex encoding to the same attributes reduced this to 4,745 pairs, representing a 97% reduction, while incorporating address attributes further decreased the count to 1,203 pairs, achieving a 99.3% reduction overall. This demonstrates that using more attributes in blocking keys significantly reduces the number of candidate pairs requiring detailed comparison.

Fig. 13 demonstrates the linear relationship between window size and complexity. A window size of 2 produced 482 pairs, increasing to 2,410 pairs at window size 5 (5× increase) and 4,820 pairs at window size 10 (10× increase). This linear scaling enables practitioners to predict computational cost when tuning recall-precision trade-offs.

Fig. 14 compares indexing approaches. Standard blocking generated only 63 pairs, achieving highest precision but lowest recall, whereas SNM with window size 10 produced 4,745 pairs–75× more comparisons, which yielded higher recall at the cost of increased computational complexity. These results demonstrate that practitioners can tune system performance through three independent parameters: blocking keys composition (achieving up to 99.3% reduction in candidate pairs), window size (with predictable linear scaling), and indexing method selection (with up to 75× difference in comparison count).

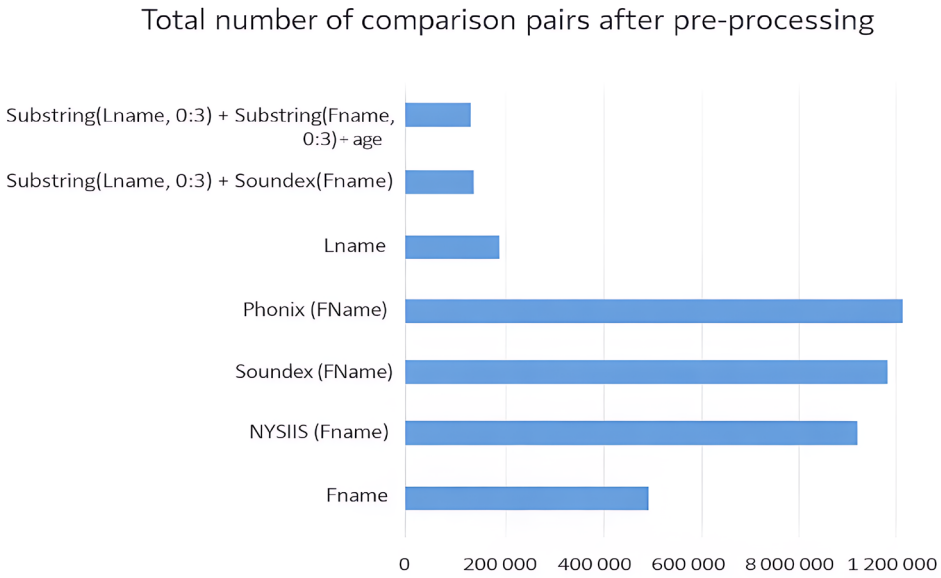

Fig. 12 – *The importance of the key selection.*

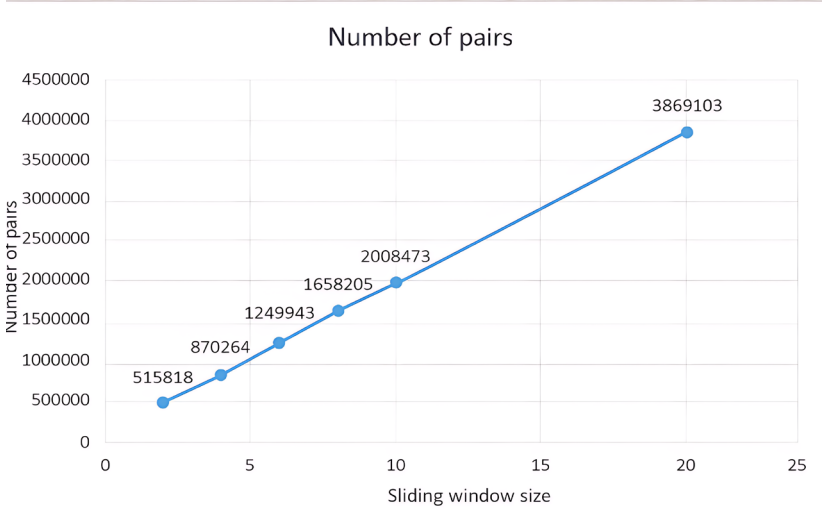

Fig. 13 –*The importance of the sliding window selection.*

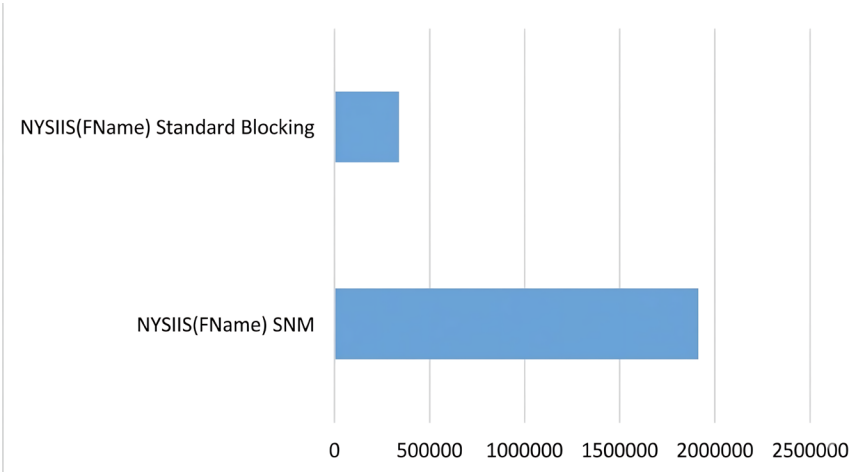

Fig. 14 – *The impact of indexing approach.*

3.8 Data updates impact

We analyzed the impact of data updates on ER results. **Table 9**, second column, describes the changes. The experiment includes some of the possible errors in data that were described earlier. Some algorithms handle specific noise better than others. Fig. 6 shows that algorithms based on Levenshtein distance have higher precision but lower recall compared to the Jaro-Winkler algorithm, which better handles phonetic variations. In the real-world system environment, the monitoring system detected 15% of errors that were previously unnoticed, demonstrating the effectiveness of the proposed method.

4 Implementation of Error Detection and Monitoring System

Our case study uses a database with millions of records. ER groups records by entity since no explicit entity identifier exists. The data are noisy and dynamic. Daily ER execution misses many links, making errors and quality issues unmeasurable without monitoring. The solution consists of two applications: a Python-based measurements and metrics calculator using the Pandas library, and a web UI. The calculator processes ER results to compute quality metrics, detect errors via domain constraints, compare consecutive outputs, and identify causes such as new, updated, or deleted records. It applies Damerau-Levenshtein for typographical corrections, stores results in JSON, and runs daily after ER execution. The Web UI visualizes results, provides real-time error alerts, and supports historical monitoring and detailed analysis.

Table 9
The impact of the data updates on ER results.

ER Algorithm	Changes in data	Classified duplicates	# of splits	FP	# of unique entities
Exact	Initial state	[[1, 2, 3], [4, 5], [6, 7]]	0	0	4
	Phonetic error Alex/Aleks	[[1, 2, 3], [4, 5]]	1	0	5
	Char interchange Jane/Jaen	[[1, 2, 3]]	2	0	6
Exact Soundex	Phonetic error and char interchange	[[1, 2, 3, 4, 5], [6, 7]]	0	1	3
Levens htein t=0.7	Phonetic error	[[1, 2, 3], [4, 5]]	1	0	5
	Chars interchange	[[1, 2, 3]]	2	0	6
Bag distance 0.8	Char is changed with the other char, typographical error	[[2, 3], [4, 5]]	2	0	6
Jaro 0.8	Char is changed with the other char	[[1, 2, 3], [4, 5]]	1	0	5
Levens htein t=0.7	Char is changed with the other char	[[1, 2, 3]]	2	0	6
	Initial data				
Levens htein t=0.7	Char deletion Jane/Jae	[[1, 2, 3], [4, 5], [6, 7]]	0	0	4
Jaro 0.8	Char deletion Jane/Jae	[[1, 2, 3], [4, 5], [6, 7]]	0	0	4
Exact	Char deletion Jane/Jae	[[1, 2, 3], [6, 7]]	1		5
Exact Soundex	Char deletion Jane/Jae	[[1, 2, 3, 5], [6, 7]]	1	1	3
Jaro 0.8	Token addition Doe/ Doe-Smith	[[1, 2, 3], [6, 7]]	1	0	5
Qgram	Token addition	[[1, 2, 3], [6, 7]]	1	0	5
Contains String	Token addition	[[1, 2, 3], [4, 5], [6, 7]]	0	0	4

4.1 Future improvements

Future iterations aim to establish a connection between records even when primary identifiers differ. This requires modifying indexing keys to group similar records and increasing the window size. A record comparison step should also be implemented, applying functions like Damerau-Levenshtein and token-based string comparison. Leverage historical data as a 'safe subset' (old addresses, corrected names) to avoid false positives. Improve address formatting using external data and automatic error correction. Probabilistic matching for high-likelihood pairs requiring manual verification should be implemented.

System Robustness to Large Data Changes. The system does not explicitly handle situations where there are sudden changes in the data, such as the

introduction of new attributes, shifts in data distribution, or a sudden increase in data volume. Future research should explore adaptive training methods that can automatically adjust classification thresholds based on data changes.

Sensitivity to False Positives. While the system achieves high precision, there is still a possibility of generating false positives. These cases can lead to incorrect entity grouping, which may have negative consequences in production.

Future improvements could include advanced filtering techniques, such as combining domain-based rules with machine learning.

Scalability in Production Environments. Experiments were conducted on controlled datasets, but real-world production environments pose significantly greater challenges related to system performance. Future research should examine the efficiency of parallelization and distributed processing to improve system scalability.

Integration with Real-Time Monitoring Systems. The current implementation enables periodic result tracking but does not support real-time analysis of data changes. A potential enhancement would involve developing a system based on stream processing technologies.

Analysis of Long-Term Performance Trends. The monitoring system focuses on individual evaluation iterations but does not include an analysis of long-term performance trends. Future work could involve time-series modeling to identify patterns of performance degradation over time.

5 Conclusions

This paper proposes a production-oriented framework for evaluating ER systems in environments where ground truth data are unavailable. By combining continuous monitoring, domain constraints, and a synthetic ground truth generator calibrated to real-world data behavior, the framework enables reliable detection of false positives and false negatives under extreme class imbalance.

Experiments on large-scale production data demonstrate that systems with very high precision (0.99) may still suffer from substantial recall loss due to data updates and noise. Monitoring-driven improvements to preprocessing and blocking reduced false negatives by up to 55× and increased recall from 0.41 to 0.68 without sacrificing precision. The results show that targeted, data-driven refinement is more effective than artificial dataset rebalancing.

Future work will explore machine learning integration, real-time monitoring, and distributed processing to further enhance robustness and scalability. We developed a ground truth generator capturing real-world particularities. We investigated ER evaluation approaches theoretically and practically. We model changes using graphs; this enables entity evolution tracking. Detailed change analysis reveals data particularities. Together, these measurements constitute our

monitoring system. We have improved the ER system and can quantitatively measure the impact of future enhancements.

6 References

[1] P. Christen: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Springer, Heidelberg, New York, 2012.

[2] I. P. Fellegi, A. B. Sunter: A Theory for Record Linkage, Journal of the American Statistical Association, Vol. 64, No. 328, December 1969, pp. 1183 – 1210.

[3] M. A. Hernández, S. J. Stolfo: The Merge/Purge Problem for Large Databases, ACM SIGMOD Record, Vol. 24, No. 2, May 1995, pp. 127 – 138.

[4] P. Christen: A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication, IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 9, September 2012, pp. 1537 – 1555.

[5] A. Haug, F. Zachariassen, D. van Liempd: The Costs of Poor Data Quality, Journal of Industrial Engineering and Management, Vol. 4, No. 2, 2011, pp. 168 – 193.

[6] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, J. Widom: Swoosh: A Generic Approach to Entity Resolution, The VLDB Journal – The International Journal on Very Large Data Bases, Vol. 18, No. 1, January 2009, pp. 255 – 276.

[7] L. Charlin, R. Zemel, C. Boutilier: Active Learning for Matching Problems, Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML), Edinburgh, Scotland, June 2012, 139 – 146.

[8] S. E. Whang, H. Garcia-Molina: Entity Resolution with Evolving Rules, Proceedings of the VLDB Endowment, Vol. 3, No. 1-2, September 2010, pp. 1326 – 1337.

[9] H. Köpcke, A. Thor, E. Rahm: Evaluation of Entity Resolution Approaches on Real-World Match Problems, Proceedings of the VLDB Endowment, Vol. 3, No. 1-2, September 2010, pp. 484 – 493.

[10] D. Menestrina, S. E. Whang, H. Garcia-Molina: Evaluating Entity Resolution Results, Proceedings of the VLDB Endowment, Vol. 3, No. 1-2, September 2010, pp. 208 – 219.

[11] M. Barnes: A Practioner's Guide to Evaluating Entity Resolution Results, arXiv:1509.04238v1 [cs.DB], September 2015, pp. 1 – 6.

[12] J. A. Hammerton, M. Granitzer, D. Harvey, M. Hristakeva, K. Jack: On Generating Large-scale Ground Truth Datasets for the Deduplication of Bibliographic Records, Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS), Craiova, Romania, June 2012, pp. 1 – 12.

[13] S. E. Whang, O. Benjelloun, H. Garcia-Molina: Generic Entity Resolution with Negative Rules, The VLDB Journal – The International Journal on Very Large Data Bases, Vol. 18, No. 6, December 2009, pp. 1261 – 1277.

[14] V. Efthymiou, K. Stefanidis, V. Christophides: Big Data Entity Resolution: From Highly to Somehow Similar Entity Descriptions in the Web, Proceedings of the IEEE International Conference on Big Data, Santa Clara, USA, October 2015, pp. 401 – 410.

[15] F. M. Naini, J. Unnikrishnan, P. Thiran, M. Vetterli: Where You Are Is Who You Are: User Identification by Matching Statistics, arXiv:1512.02896v1 [cs.LG], December 2015, pp. 1 – 14.

[16] L. Zhang, R. Vaisenberg, S. Mehrotra, D. V. Kalashnikov: Video Entity Resolution: Applying ER Techniques for Smart Video Surveillance, Proceedings of the IEEE International

Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Seattle, USA, March 2011, pp. 26 – 31.

[17] R. Al-Kamha, D. W. Embley: Grouping Search-engine Returned Citations for Person-name Queries, Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Management (WIDM), Washington, USA, November, 2004, pp. 93 – 103.

[18] D. Bharambe, S. Jain, A. Jain: A Survey: Detection of Duplicate Record, International Journal of Emerging Technology and Advanced Engineering, Vol. 2, No. 11, November 2012, pp. 298 – 307.

[19] E. QAS: Exploiting the Single Customer View to Maximise the Value of Customer Relationships, Experian QAS, 2012.

[20] V. Efthymiou, K. Stefanidis, V. Christophides: Minoan ER: Progressive Entity Resolution in the Web of Data, Proceedings of the 19th International Conference on Extending Database Technology (EDBT), Bordeaux, France, March 2016, pp. 370 – 371.

[21] V. Christophides, V. Efthymiou, K. Stefanidis: Entity Resolution in the Web of Data – Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool, Palo Alto, 2015.

[22] J. Jonas: Threat and Fraud Intelligence, Las Vegas Style, IEEE Security & Privacy, Vol. 4, No. 6, November 2006, pp. 28 – 34.

[23] S. Bartunov, A. Korshunov, S.- T. Park, W. Ryu, H. Lee: Joint Link-attribute User Identity Resolution in Online Social Networks, Proceedings of the 6th SNA-KDD Workshop, Beijing, China, August 2012, pp.1 – 9.

[24] M. A. Jaro: Probabilistic Linkage of Large Public Health Data Files, Statistics in Medicine, Vol. 14, No. 5-7, March-April 1995, pp. 491 – 498.

[25] L. F. Carvalho, A. H. F. Laender, W. Meira Jr.: Entity Matching: A Case Study in the Medical Domain, Proceedings of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW), Lima, Peru, May 2015, pp. 1 – 12.

[26] H. Köpcke, A. Thor, S. Thomas, E. Rahm: Tailoring Entity Resolution for Matching Product Offers, Proceedings of the 15th International Conference on Extending Database Technology (EDBT), Berlin, Germany, March 2012, pp. 545 – 550.

[27] T. N. Herzog, F. J. Scheuren, W. E. Winkler: Data Quality and Record Linkage Technique, Springer, New York, 2007.

[28] S. E. Whang, D. Menestrina, G. Koutrika, M. Theobald, H. Garcia-Molina: Entity Resolution with Iterative Blocking, Proceedings of the ACM SIGMOD International Conference on Management of Data, Providence Rhode Island, USA, June 2009, pp. 219 – 232.

[29] P. Christen, D. Vatsalan, Z. Fu: Advanced Record Linkage Methods and Privacy Aspects for Population Reconstruction – A Survey and Case Studies, Ch. 5, Population Reconstruction, Springer, Cham, New York, London, 2015.

[30] I. Bhattacharya, L. Getoor: Collective Entity Resolution in Relational Data, ACM Transactions on Knowledge Discovery from Data (TKDD), Vol. 1, No. 1, March 2007, p. 5.

[31] P. Christen, K. Goiser: Quality and Complexity Measures for Data Linkage and Deduplication, Ch. 6, Quality Measures in Data Mining, Springer, Berlin Heidelberg, 2007.

[32] T. Bachteler, J. Reiher: TDGen: A Test Data Generator for Evaluating Record Linkage Methods, German Record Linkage Center, No. WP-GRLC-2012-01, July 2012, pp. 1 – 16.

[33] P. Christen: Febrl – A Freely Available Record Linkage System with a Graphical User Interface, Proceedings of the 2nd Australasian Workshop on Health Data and Knowledge Management, Wollongong, Australia, January 2008, pp. 17 – 25.

[34] S. E. Whang, H. Garcia-Molina: Incremental Entity Resolution on Rules and Data, The VLDB Journal – The International Journal on Very Large Data Bases, Vol. 23, No. 1, February 2014, pp. 77 – 102.

[35] V. Levensthein: Binary Codes Capable of Correcting Deletions, Insertions and Reversals, Soviet Physics – Doklady, Vol. 10, No. 8, February 1966, pp. 707 – 710.

[36] F. J. Damerau: A Technique for Computer Detection and Correction of Spelling Errors, Communications of the ACM, Vol. 7, No. 3, March 1964, pp. 171 – 176.

[37] A. Z. Broder, S. C. Glassman, M. S. Manasse, G. Zweig: Syntactic Clustering of the Web, Computer Networks and ISDN Systems, Vol. 29, No. 8-13, September 1997, pp. 1157 – 1166.

[38] J. R. Wang, S. E. Madnick: The Inter-Database Instance Identification Problem in Integrating Autonomous Systems, Proceedings of the 5th IEEE International Conference on Data Engineering (ICDE), Los Angeles, USA, February 1989, pp. 46 – 55.

[39] S. Tejada, C. A. Knoblock, S. Minton: Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification, Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, July 2002, pp. 350 – 359.

[40] R. Kohavi, F. Prohost: Glossary of Terms, Machine Learning, Vol. 30, No. 2-3, February 1998, pp. 271 – 274.

[41] A. Solomonoff, A. Mielke, M. Schmidt, H. Gish: Clustering Speakers by their Voice, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seattle, USA, May 1998, pp. 757 – 760.

[42] J. Ajmera, H. Bourlard, I. Lapidot: Improved Unknown Multiple Speaker Clustering Using HMM, Technical Repeport, IDIAP, 2002.

[43] A. Doan, Y. Lu, Y. Lee, J. Han: Profile-Based Object Matching for Information Integration, IEEE Intelligent Systems, September/October 2003, pp. 54 – 59.

[44] S. Guo, X. L. Dong, D. Srivastava, R. Zajac: Record Linkage with Uniqueness Constraints and Erroneous Values, Proceedings of the VLDB Endowment, Vol. 3, No. 1-2, September 2010, pp. 417 – 428.

[45] Z. Fu, P. Christen, J. Zhou: A Graph Matching Method for Historical Census Household Linkage, Proceedings of the 18th Pacific-Asia Conference (PAKDD), Tainan, Taiwan, May 2014, pp. 458 – 496.

[46] B. Li: Entity Resolution over Graphs, M.Sc. Thesis, Australian National University, Canberra, 2014.

[47] H. Kardes, D. Konidena, S. Agrawal, M. Huff, A. Sun: Graph-based Approaches for Organization Entity Resolution in MapReduce, Proceedings of the TextGraphs-8 Workshop, Seattle, USA, October 2013, pp. 70 – 78.

[48] K. A. Robbins, C. Jeffery, S. Robbins: Visualization of Splitting and Merging Processes, Journal of Visual Languages and Computing, Vol. 11, No. 6, December 2000, pp. 593-614.

[49] L. Kolb, E. Rahm: Parallel Entity Resolution with Dedoop, Datenbank-Spektrum, Vol. 13, No. 1, March 2013, pp. 23 – 32.

[50] H. Li, L. Feng, S. Li, F. Hao, C. J. Zhang, Y. Song: On Leveraging Large Language Models for Enhancing Entity Resolution: A Cost-Efficient Approach, arXiv:2401.03426v2 [cs.CL], September 2024, pp. 1 – 9.

[51] M. Berrendorf, E. Faerman, V. Melnychuk, V. Tresp, T. Seidl: Knowledge Graph Entity Alignment with Graph Convolutional Networks: Lessons Learned, Proceedings of the 42nd European Conference on IR Research (ECIR), Lisbon, Portugal, April 2020, pp. 3 – 11.

[52] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra: Deep Learning for Entity Matching: A Design Space Exploration, Proceedings of the ACM SIGMOD International Conference on Management of Data, Houston, USA, June 2018, pp. 19 – 34.

[53] Y. Li, J. Li, Y. Suhara, A. Doan, W.- C. Tan: Deep Entity Matching with Pre-trained Language Models, Proceedings of the VLDB Endowment, Vol. 14, No. 1, September 2020, pp. 50 – 60.

[54] R. Peeters, A. Steiner, C. Bizer: Entity Matching Using Large Language Models, arXiv:2310.11244v4 [cs.CL], October 2024, pp. 1 – 13.