PyResolveMetrics: A Standards-Compliant and Efficient Approach to Entity Resolution Metrics

Andrei Olar[1][a]
Laura Dioşan[1][b]
[1] *Faculty of Mathematics and Computer Science, Babeş-Bolyai University*
andrei.olar@ubbcluj.ro, laura.diosan@ubbcluj.ro

Keywords: Entity Resolution,Metrics,Library,Open Source,Standards-Compliant,Theoretical Model,Efficiency

Abstract: Entity resolution, the process of discerning whether multiple data refer to the same real-world entity, is crucial across various domains, including education. Its quality assessment is vital due to the extensive practical applications in fields such as analytics, personalized learning or academic integrity. With Python emerging as the predominant programming language in these areas, this paper attempts to fill in a gap when evaluating the qualitative performance of entity resolution tasks by proposing a novel consistent library dedicated exclusively for this purpose. This library not only facilitates precise evaluation but also aligns with contemporary research and application trends, making it a significant tool for practitioners and researchers in the field.

1 Introduction

The field of entity resolution (ER), integral to natural language processing and emerging technologies in education, is pivotal in understanding and linking data across multiple sources. Some definitions view it as identifying and linking data from multiple sources [Qian et al., 2017]. However, it's argued that this identification and linking is a more specialized process [Talburt, 2011].

ER, also known as record linkage, data deduplication, merge-purge, named entity recognition, entity alignment, and entity matching, plays a significant role in educational communication and collaboration tools, facilitating the information exchange between parents, teachers and students. It can also be useful to track the progression of alumni, by linking various data sources to provide meaningful insights on the career trajectories of graduates. It is a key component in AI literacy, particularly in those machine learning tasks that involve identifying how disparate pieces of information correlate to the same real-world entity. ER itself has numerous implementations that rely on machine learning and is important for AI literacy for that reason, too [Li et al., 2020]. We recognize the importance of accurate and efficient ER in both individual learning outcomes and broader societal impacts. Enhancing research integrity through plagiarism detection or enabling personalized learning by tracking a student's preferences across learning domains might qualify as fields of study in their own right. For example, the work of [Chen et al., 2021] points out the importance of University and professional information for career exploration. Their survey highlights that matching the educational offering to fit student goals leads to a higher chance of students developing successful careers. One can envision systems that automatically create educational offerings based on the profiles of students. In this scenario, ER has the role of automatically building the student profile from heterogeneous information sources. Another use case for automatically generated profiles might be career recommender systems, for example. The ideal outcome from using the information stored in these profiles would be finding the best possible match between educational offering and student aspirations.

What if ER provides us with a misleading profile? At best, we realize this is the case, stop trusting the ER system and revert to a state where we don't benefit from the information stored in the profiles generated through ER. At worst, we do not realize the system error and proceed career exploration based on misleading profiles. This leads to bad career choice recommendation and more severe risks related to wasted time, financial misfortune, professional dissatisfaction, stagnation in personal development or even health and relationship concerns. Adopting the

[a] https://orcid.org/0009-0006-7913-9276
[b] https://orcid.org/0000-0002-6339-1622

right ER system is desirable for obtaining the initial benefits of making more informed decisions faster. ER systems should not be adopted and cannot be maintained without measuring the quality of their outcomes.

In this context, the paper introduces a new Open Source library that is hosted in a Git repository on GitHub [PyResolveMetrics, 2023]. The library offers implementations of well known metrics for evaluating ER, contributing significantly to metrics and evaluation in educational technologies. Thus, this library could have a role in advancing tools and methodologies in the realm of education and technology by allowing a more informed process for developing tools that make use of ER. The book [Schütze et al., 2008] studies various methods for evaluating the performance of information retrieval systems that help in assessing how effective these systems are in searching, identifying, and retrieving relevant information from large datasets. The metrics revolve around the notion of relevant and irrelevant information that is retrieved by the system. It is asserted that what is relevant is stipulated in a ground truth which is dependant upon an information need. ER systems partly function as information retrieval systems, as they determine whether multiple data points refer to the same real-world entity. This capability to discern data identity within a context is the fundamental information need of any ER system. Is it therefore fitting to use information retrieval metrics for entity resolution? This seems to be the consensus drawn in the scientific literature as we shall see in the next section. Our library implements various information retrieval metrics adapted for ER.

It's also important to acknowledge the distinct entity resolution models. The library sets itself apart by organizing metrics based on their compatibility with ER models, influenced by the underlying differences in data structures that are characteristic to each model. Special attention is given to interoperability and the seamless integration of the library into the Python programming language ecosystem. Its key features are: embracing an OpenSource licensing model, efficient implementation using state of the art libraries under a very popular platform, and a design that is agnostic to the ER implementation.

After this introduction, we overview two existing mathematical models for ER which are widely used and still represent the state of the art. Then we go through other work that relates to this paper. Subsequently, we present the new library and pay special attention to the reasons for implementing it. We go over the metrics that are implemented, the technological and design choices that were made, an example of using the library and a performance evaluation of the functions implemented by the library. In the end we offer some conclusions and present aspects that require more work.

2 Entity Resolution Models

Fellegi-Sunter Model. In the late 1960s Ivan Fellegi and Alan Sunter wrote the seminal paper [Fellegi and Sunter, 1969] for record linkage — what would later be known as ER. To this day, their mathematical model based on probability theory is the most popular way of formalizing the ER problem. In this mathematical model, ER is a function that aids in probabilistic decision making. In this model of ER, the process primarily involves comparing data from two sources. The essential step is matching two items — one from each source, after which a decision is made to categorize the match as a 'link', 'non-link', or 'possible link'. Consequently, any matching algorithm under this model typically returns pairs of items from the original data sources, each tagged as one of these categories. However, in practical applications today, this process is often simplified to just returning a list of pairs labeled as 'links'. This intuitive explanation gives us the structure of the input we can expect when we use the Fellegi-Sunter ER model: an iterable sequence of pairs.

The metrics that are implemented with the Fellegi-Sunter model of entity resolution in mind will accept iterable sequences of pairs as input where the ground truth and the result of the ER task are concerned.

Algebraic Model. The algebraic model for ER, initially conceived for assessing information quality in large datasets [Talburt et al., 2007], was later refined to describe the ER process itself [Talburt, 2011]. This model treats ER as an algebraic equivalence relation over a given input set, which can include data from as many original sources as necessary. The unique aspect of this model lies in the characteristics of equivalence relations [Halmos, 1960]. These relations create partitions over the input set, with each partition component equivalent to an equivalence class of the relation [Talburt, 2011]. Conversely, a partition over a set can also induce an equivalence relation. With this in mind, evaluating the outcome of an ER task becomes as easy as comparing two partitions: the partition that induces the ideal equivalence relation (the gold standard or ground truth) to the partition that is produced by the ER task.

The library supports a few metrics for comparing partitions, all of which expect that a partition is represented as a list of sets.

3 Related Work

Measuring ER quality was a subject of interest ever since the first paper on the subject surfaced [Newcombe et al., 1959]. It speaks about accuracy and contamination similarly to the current notions of true and false positives. The fundamental theory of record linkage [Fellegi and Sunter, 1969] offers a probabilistic approach to evaluating the success of an ER task. It suggests methods to affect the results through the selection of suitable thresholds for defining success and failure. It also provides mechanisms for properly weighting for independent probability variables. The literature expands on these techniques in subsequent papers [Winkler, 1990]. Some of the ER evaluation metrics that are a direct result of this theoretical foundation include match accuracy, match rate [Jaro, 1989], error rate estimation, rate of clerical disambiguation [Winkler, 1990] or relative distinguishing power of matching variables [Winkler, 2014]. A lot of effort is spent on estimating and measuring the effectiveness of blocking techniques to reduce the input size of the data set used for evaluation purposes [Winkler, 1990, Jaro, 1989]. Measuring ER performance was and remains a computationally intensive task.

Concerns about using accuracy and match rate are also voiced [Goga et al., 2015]. Thus we see a shift towards metrics used in the related field of information retrieval. The probabilistic model for ER aligns well with concepts such as true/false positives/negatives. Given the extensive history of using ground truths to assess entity resolution quality, there is a natural fit for using information retrieval quality metrics. Most literature on this topic focuses on using information retrieval metrics where the order in which results are retrieved is not relevant [Schütze et al., 2008].

Besides the original statistical model for ER, other models have evolved from it or alongside it. The work of the InfoLab at Stanford on their Stanford Entity Resolution Framework [Benjelloun et al., 2009] and that of the Center for Entity Resolution and Information Quality at the University of Arkansas in Little Rock [Talburt et al., 2007] stand out. These models of ER also propose new metrics for evaluating ER quality [Menestrina et al., 2010, Talburt, 2011].

There is ample coverage of the metrics used for ER in syntheses on the subject [Köpcke et al., 2010, Maidasani et al., 2012, Talburt, 2011]. Clustering metrics such as pairwise and cluster metrics [Menestrina et al., 2010, Huang et al., 2006] or the Rand index [Talburt et al., 2007] seem to be used more frequently to measure ER quality as time passes.

Numerous systems to perform ER are available. Some of them include modules to evaluate the performance of a particular ER solution [Köpcke et al., 2009, Doan et al., 2020, University of Arkansas Little Rock, 2012]. There are also other Python packages that implement some or all of the metrics provided by our library [Virtanen et al., 2020, paulboosz, 2018].

4 PyResolveMetrics

In this context, the necessity for yet another specialized library dedicated to evaluating ER metrics might seem redundant. This skepticism is rooted in the expectation that Python, being a highly popular programming platform, should already offer high-quality, reusable tools available for a wide range of applications — including evaluating ER results.

Upon closer examination of the tools available for evaluating entity resolution tasks, certain limitations in the existing assumptions become apparent. There are indeed numerous libraries offering packages for computing entity resolution metrics. However, using a general-purpose library like SciPy raises concerns about interoperability and efficiency. This is particularly relevant when the sole requirement is to compute entity resolution metrics, and the additional features of a comprehensive library are unnecessary. The challenge of seamlessly integrating ER evaluation routines into a custom built project becomes even more pronounced when attempting to use the ones packaged with established ER systems [University of Arkansas Little Rock, 2012], [Papadakis et al., 2017], [Li et al., 2020], [Doan et al., 2020].

Conversely, when specifically searching for libraries that only offer ER metrics, it becomes evident that some of the essentials for effectively evaluating ER tasks may be absent [paulboosz, 2018].

Approaching the issue from a different angle, using metrics from a general-purpose algorithmic library like Scipy (specifically `scipy.metrics`) for ER evaluation requires strict adherence to certain design choices imposed by the library. For example, to calculate the Rand index, data clusters must be mapped with labels, and these labels must be provided as input. While this might seem simple, the user-friendliness of such an approach is debatable. The complexity of adapting existing data and managing the necessary labels for the package could potentially rival the complexity of computing the Rand index itself, mooting the use of the package. Furthermore, additional memory and compute time are also required to perform the mapping between own data structures and the ones required by the API contract.

In short, here are the reasons we chose to implement such a library:

- Architecturally, adhering to the principle of 'do one thing and do it well' is beneficial. This approach avoids the biases and dependencies of general-purpose libraries like SciPy, which can complicate integration into our custom-designed software.
- Historically, ER has adapted evaluation techniques from statistics, information retrieval, and graph theory, tailoring these methods to suit its specific needs. It seems desirable to standardize these methods into forms specific to ER.
- Currently, there appears to be no implementation that consolidates all the metrics useful for ER evaluation, as identified in scientific literature, into a single, cohesive unit.
- Our work has a significant component of evaluating ER outcomes.

Our opinion is that using mathematical models specific to ER is the best approach for guiding the library's design. Since each model significantly impacts the data structures used in evaluation, the library's functions are categorized based on the type of input they support and, implicitly, by the mathematical model they align with. There are a couple of important assumptions that the library makes, regardless of the ER model. One such assumption is that the quality of the ER output is always measured against a ground truth [Schütze et al., 2008]. The other assumption is that the ground truth and the ER result are both structured under the same mathematical model.

4.1 Supported Metrics

Statistical quality metrics, extensively detailed in the literature [Schütze et al., 2008, Maidasani et al., 2012], are the most common method for measuring ER performance as evidentiated by their almost ubiquitous usage [Köpcke et al., 2009,Goga et al., 2015,Li et al., 2020, Obraczka et al., 2021]. These metrics are linked to the Fellegi-Sunter model for ER which provides clear definitions of Type I and Type II errors [Winkler, 1990]. Type I and Type II errors clarify the concepts of true positives, true negatives, false positives, and false negatives as they are used in entity resolution. Understanding these concepts necessitates referencing the M (matches) and U (non-matches) sets as defined in the seminal paper on the model.

Depending on the expected location of a pair produced by the entity resolution function, we define:

- **true positives** as pairs predicted to be in M that should be in M,
- **false positives**, or type I errors, as pairs predicted to be in M, but should be in U,
- **true negatives** as pairs predicted to be in U that should be in U, and
- **false negatives**, or type II errors, as pairs predicted to be in U, but should be in M.

Several metrics based on these concepts exist, though the effectiveness of some has been questioned [Goga et al., 2015]. With this in mind we finally define the three quality metrics that are supported by our library:

$$Precision = \frac{true\,positives}{true\,positives + false\,positives} \tag{1}$$

$$Recall = \frac{true\,positives}{true\,positives + false\,negatives} \tag{2}$$

$$F_1 Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

Precision (or the positive predictive value) is defined as the number of correct predictions that were made in relation to the total number of predictions that were made. *Recall* (or sensitivity) is defined as the number of correct predictions that were made in relation to the total number of positive predictions that could have been made (which corresponds to the number of items in the ground truth). The F_1 score is the harmonic mean of the precision and the recall and it is used to capture the tradeoff between precision and recall [Maidasani et al., 2012].

Algebraic metrics is the generic name we use for 'cluster metrics' [Rand, 1971,Maidasani et al., 2012] and 'pairwise metrics' [Maidasani et al., 2012,Menestrina et al., 2010] because their foundation is algebraic and because they are linked to the algebraic model for ER. Most of the algebraic metrics implemented by the library are an exercise in using operations on sets, while the rest focus on matrix operations with a dash of combinatorics:

- Pairwise metrics (precision, recall and F-measure) [Menestrina et al., 2010,Maidasani et al., 2012]
- Cluster metrics (precision, recall and F-measure) [Huang et al., 2006,Maidasani et al., 2012]
- Talburt-Wang Index [Talburt et al., 2007]
- Rand [Rand, 1971] and Adjusted Rand Index [Hubert and Arabie, 1985]

Their input arguments (the ground truth and the ER result) are represented as partitions over *the same* set.

The Rand index is one of the first metrics used to compare the similarity between two different data clusterings. It quantifies the agreement or disagreement between these clusterings by considering pairs of elements.

$$RandIndex = \frac{(a+b)}{\binom{n}{2}} \tag{4}$$

The main components of the Rand index are as follows: a: Represents the number of times a pair of elements belongs to the same cluster across both clustering methods, b: Represents the number of times a pair of elements belongs to different clusters across both clustering methods, $\binom{n}{2}$: denotes the number of unordered pairs in a set of n elements.

The Rand index always takes values in the $[0,1)$ interval.

A variation on the Rand Index is the Adjusted Rand Index for chance grouping of elements. It accounts for agreements between data clusterings that occur due to chance [Yeung and Ruzzo, 2001]. The Adjusted Rand Index is calculated by using the following formula:

$$ARI = \frac{RandIndex - E}{\max(RandIndex) - E}, \tag{5}$$

where E is the expected value of the RandIndex. The Adjusted Rand index is valued in the interval $[-1,1]$. For a comprehensive understanding of the Adjusted Rand Index and its calculation, we recommend consulting the detailed and informative work presented in the study by [Warrens and van der Hoef, 2022] on the subject. For both of these indexes, higher scores indicate a closer alignment between the compared partitions.

A metric that attempts to approximate the Rand Index is the Talburt-Wang Index which counts the number of overlapping subsets of two partitions over the same input set. Assuming A and B are two partitions over the same input set of elements, the Talburt-Wang Index is given by the formula:

$$\Delta(A,B) = \frac{|A| \cdot |B|}{\Phi(A,B)^2} \tag{6}$$

where $\Phi(A,B) = \sum_{i=1}^{|A|} \{B_j \in B | B_j \cap A_i \neq \emptyset\}$.

This metric approximates the Rand Index without requiring the expensive counting of true positives, false positives, true negatives or false negatives [Talburt et al., 2007]. It is valued within the same interval as the Rand Index.

Our library implements other popular metrics that can be used for comparing partitions: pairwise precision, pairwise recall and their harmonic mean (the pairwise F_1 measure) [Maidasani et al., 2012].

If we have two sets X and Y, the pairwise precision is given by the ratio of pairs that are in both sets to the total amount of pairs of the reference set.

$$PP(X,Y) = \frac{|Pairs(X) \cap Pairs(Y)|}{|Pairs(X)|} \tag{7}$$

The pairwise recall is given by the ratio of pairs that are in both sets to the number of pairs in the comparison set [Maidasani et al., 2012].

$$PR(X,Y) = \frac{|Pairs(X) \cap Pairs(Y)|}{|Pairs(Y)|} \tag{8}$$

The pairwise F-measure is given by the harmonic mean of the pairwise precision and pairwise recall.

$$PF = \frac{2 \cdot PP \cdot PR}{PP + PR} \tag{9}$$

The library computes partition metrics by iteratively analyzing equivalence classes within each partition generated by the ER equivalence relation and extracting element pairs from each subset.

Finally, our library supports ‘cluster measures’ [Maidasani et al., 2012]. Cluster precision is the ratio of the number of completely correct clusters to the total number of clusters resolved, whereas cluster recall is the portion of true clusters resolved [Huang et al., 2006]. The harmonic mean of the cluster precision and cluster recall is typically called the cluster F-measure. In this paragraph ‘clusters’ refer to the equivalence classes of the entity resolution relation as it is formalized in the algebraic model.

Given two partitions A and B, the cluster measures are given by the following formulae:

$$CP(A,Y) = \frac{|A \cap B|}{|A|} \tag{10}$$

$$CR(A,Y) = \frac{|A \cap B|}{|B|} \tag{11}$$

$$CF = \frac{2 \cdot CP \cdot CR}{CP + CR} \tag{12}$$

4.2 Technology

The technology used for implementing our library is described in Section 1 of the Appendix available online.

4.3 Example Usage

To provide a visual outlook over the metrics provided by our library we are using a toy data set [Olar, 2023] containing near duplicates and the PPJoin [Xiao et al., 2011] entity matching algorithm to perform ER. The PPJoin algorithm matches items by using prefix lengths determined using a Jaccard coefficient t.

We split the data in the toy data set into two data sets by column. The resulting data sets are:

DG1: with `'name'`, `'manufacturer'`, `'price'`, `'id'`, and

DG2: with `'description'`, `'name'`, `'id'`

Because we have split the data column-wise, we know exactly what the ground truth should be for each of the metrics, assuming that each row in the original toy data set refers to a distinct real-world entity. Because we are working with two data sets, the ground truth for the statistical model is the same as the ground truth for the algebraic model: a list of pairs of matching items obtained by iterating over DG1 and DG2 using the same cursor.

All that's left is to apply the PPJoin algorithm on DG1 and DG2 and plot the values of the metrics provided by the library for values of t in the interval $[0,1)$ at increments of 0.01. The plots that show the outcome are available online in Section 2 of the accompanying Appendix.

4.4 Performance

CPU performance is usually evaluated by throughput (e.g millions of operations per second or MIPS). However it is meaningless to compare throughput on different CPU architectures [Jain, 1991].

A similar concern can be raised about memory profiling in relation to the underlying operating system. Due to the variability of the outcomes during experimentation and the fact that all memory consumption is very dependent on the operating system and standard C library used for compiling and linking the Python interpreter, we found memory profiling not to provide great insights.

Under these circumstances we have chosen to elaborate a method of profiling the CPU usage of the library which is agnostic to the underlying hardware. CPU profiling is useful in the context of judging the metrics provided by the library relatively to one another.

To prevent the ER task from interfering with profiling the metrics library, we run our experiment in two stages. In the first stage we run the ER task and store its result in a file along with the ground truth. The second stage loads the results and ground truth from the file and runs the entity resolution metrics while profiling the CPU usage.

The performance analysis is available online in Section 3 of the Appendix to this article. Perhaps the most important lesson to learn from profiling our library is that algebraic metrics are an order of magnitude more expensive to compute than statistical metrics. Moreover, not all algebraic metrics were created equal: the Rand indexes are an order of magnitude more expensive to compute than the other algebraic metrics. Therefore, because the Talburt-Wang index approximate the Rand index well, it might be a preferable choice to measure how well an ER algorithm performs clustering.

5 Conclusions and Future Work

We have introduced a library for evaluating ER results that is based on standards and Python protocols, making it highly interoperable. The API exposed by this library is deeply rooted in the mathematical models fundamental to ER, making it more familiar to ER users.

The performance of the library is sound because it externalizes computationally expensive tasks to native code. The accuracy of the implemented metrics is verified automatically through unit tests.

These attributes render the library not only highly beneficial but also low maintenance, making it an invaluable asset ER. This does not preclude additional work.

Missing metrics. The ER models we have touched upon support many more metrics that the library does not currently implement. The work by [Maidasani et al., 2012] provides an insightful overview. For well-rounded support of the ER models mentioned so far, the library should implement at least additional cluster comparisons, such as the Closest Cluster F_1, the MUC F_1, B^3F_1 and the $CEAFF_1$ [Maidasani et al., 2012], and additional Rand-like indexes [Warrens and van der Hoef, 2022].

Missing models. Besides the models we have covered herein, ER has been theorized to be a graph problem [Obraczka et al., 2021] or an exercise in lattice theory with an ordering relation based on "merge dominance" [Benjelloun et al., 2009]. More work is required to distil the metrics that become available for evaluating ER under those models and the data structures that are used in the evaluation process.

REFERENCES

[Benjelloun et al., 2009] Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E., and Widom, J. (2009). Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18:255–276.

[Chen et al., 2021] Chen, H., Liu, F., Wen, Y., Ling, L., Chen, S., Ling, H., and Gu, X. (2021). Career exploration of high school students: Status quo, challenges, and coping model. *Frontiers in Psychology*, 12.

[Doan et al., 2020] Doan, A., Konda, P., Suganthan GC, P., Govind, Y., Paulsen, D., Chandrasekhar, K., Martinkus, P., and Christie, M. (2020). Magellan: toward building ecosystems of entity matching solutions. *Communications of the ACM*, 63(8):83–91.

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

[Goga et al., 2015] Goga, O., Loiseau, P., Sommer, R., Teixeira, R., and Gummandi, K. P. (2015). On the reliability of profile matching across large online social networks. In *On the Reliability of Profile Matching Across Large Online Social Networks*, Sydney.

[Halmos, 1960] Halmos, P. R. (1960). *Naive set theory*. van Nostrand.

[Huang et al., 2006] Huang, J., Ertekin, S., and Giles, C. L. (2006). Efficient name disambiguation for large-scale databases. In *European conference on principles of data mining and knowledge discovery*, pages 536–544. Springer.

[Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2:193–218.

[Jain, 1991] Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, volume 1. Wiley New York, 1 edition.

[Jaro, 1989] Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

[Köpcke et al., 2009] Köpcke, H., Thor, A., and Rahm, E. (2009). Comparative evaluation of entity resolution approaches with fever. *Proceedings of the VLDB Endowment*, 2(2):1574–1577.

[Köpcke et al., 2010] Köpcke, H., Thor, A., and Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2):484–493.

[Li et al., 2020] Li, Y., Li, J., Suhara, Y., Doan, A., and Tan, W.-C. (2020). Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*.

[Maidasani et al., 2012] Maidasani, H., Namata, G., Huang, B., and Getoor, L. (2012). Entity resolution evaluation measures. *University of Maryland, Tech. Rep*.

[Menestrina et al., 2010] Menestrina, D., Whang, S. E., and Garcia-Molina, H. (2010). Evaluating entity resolution results. In *Evaluating entity resolution results*.

[Newcombe et al., 1959] Newcombe, H. B., Kennedy, J. M., Axford, S. J., and James, A. P. (1959). Automatic linkage of vital records. *Science*, 130(3381):954–959.

[Obraczka et al., 2021] Obraczka, D., Schuchart, J., and Rahm, E. (2021). Eager: embedding-assisted entity resolution for knowledge graphs. *arXiv preprint arXiv:2101.06126*.

[Olar, 2023] Olar, A. (2023). Experiment data. https://github.com/matchescu/experiment-data. Online; accessed 25.10.2023.

[Papadakis et al., 2017] Papadakis, G., Tsekouras, L., Thanos, E., Giannakopoulos, G., Palpanas, T., and Koubarakis, M. (2017). Jedai: The force behind entity resolution. In *The Semantic Web: ESWC 2017 Satellite Events: ESWC 2017 Satellite Events, Portorož, Slovenia, May 28–June 1, 2017, Revised Selected Papers 14*, pages 161–166. Springer.

[paulboosz, 2018] paulboosz (2018). entity-resolution-evaluation. https://github.com/entrepreneur-interet-general/entity-resolution-evaluation/README.md. accessed 2023-09-22.

[PyResolveMetrics, 2023] PyResolveMetrics (2023). Pyresolvemetrics. https://github.com/matchescu/er-metrics. Online; Accessed: 26.11.2023.

[Qian et al., 2017] Qian, K., Popa, L., and Prithviraj, S. (2017). Active learning for large-scale entity resolution. In *Active learning for large-scale entity resolution*, pages 1379–1388.

[Rand, 1971] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

[Schütze et al., 2008] Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

[Talburt et al., 2007] Talburt, J., Wang, R., Hess, K., and Kuo, E. (2007). An algebraic approach to data quality metrics for entity resolution over large datasets. In *Information quality management: Theory and applications*, pages 1–22. IGI Global.

[Talburt, 2011] Talburt, J. R. (2011). *Entity resolution and information quality*. Elsevier.

[University of Arkansas Little Rock, 2012] University of Arkansas Little Rock, E. (2012). Oyster. https://bitbucket.org/oysterer/oyster/src/master/README.md. accessed 2023-09-22.

[Virtanen et al., 2020] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

[Warrens and van der Hoef, 2022] Warrens, M. J. and van der Hoef, H. (2022). Understanding the adjusted rand index and other partition comparison indices based on counting object pairs. *Journal of Classification*, 39(3):487–509.

[Winkler, 1990] Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Non-Journal*.

[Winkler, 2014] Winkler, W. E. (2014). Matching and record linkage. *WIREs Computational Statistics*, 6(5):313–325.

[Xiao et al., 2011] Xiao, C., Wang, W., Lin, X., Yu, J. X., and Wang, G. (2011). Efficient similarity joins for near-duplicate detection. *ACM Transactions on Database Systems (TODS)*, 36(3):1–41.

[Yeung and Ruzzo, 2001] Yeung, K. Y. and Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774.