A Review of Record Linkage Methods: Similarity Scoring, Matching Strategies, and Evaluation

Huang Hongnan, Li Zhengguang, Wu Gefei
Sichuan University
Email: 2022141520186@stu.scu.edu.cn

***Abstract*—Record linkage, also known as entity resolution or duplicate detection, is the process of identifying records that refer to the same real-world entity across different datasets. This paper reviews foundational and modern approaches to record linkage, focusing on similarity scoring techniques, matching decision strategies, and evaluation metrics. We discuss the use of character-based, token-based, and phonetic similarity functions, thresholding methods including the Fellegi-Sunter model, and supervised classification frameworks. We also highlight best practices in evaluating linkage performance, emphasizing precision, recall, and F1-score over traditional accuracy. This review aims to provide a practical and theoretical foundation for developing robust linkage systems.**

***Index Terms*—record linkage, entity resolution, similarity scoring, deduplication, F1-score, supervised classification**

I. Introduction

Record linkage (RL) refers to identifying and unifying records referring to the same entity across disparate data sources [1]. It is essential in data integration, national statistics, healthcare, and any domain involving duplicate or inconsistent entries. Traditional approaches focused on rule-based systems, while modern methods emphasize statistical and learning-based techniques [2], [3].

Challenges include typographical errors, format inconsistencies, and missing data [4]. Our HW1 task, involving *deduplication* and *canonical mapping*, illustrates the importance of accurate similarity scoring and matching strategies in practical scenarios.

II. Overview of the Record Linkage Process

Although details can vary by project, the record linkage process is commonly divided into five main stages [2], [5], [6]. Figure 1 provides a visual summary of these stages, illustrating a typical end-to-end workflow from data preprocessing to final evaluation.

A. Preprocessing

Goal: Clean and standardize data to ensure reliable data quality.
Methods: Removing extraneous characters, normalizing name formats (e.g., “Dr.” → “Doctor”), handling missing values, applying domain-specific rules, etc.
Relevance to HW1: Ensuring all name fields from *primary*, *alternate*, and *test* files are normalized before any comparisons helps reduce errors.

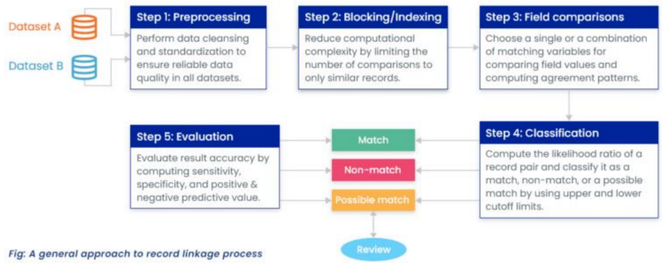
Dataset A
Dataset B
Step 1: Preprocessing
Perform data cleansing and standardization to ensure reliable data quality in all datasets.
Step 2: Blocking/Indexing
Reduce computational complexity by limiting the number of comparisons to only similar records.
Step 3: Field comparisons
Choose a single or a combination of matching variables for comparing field values and computing agreement patterns.
Step 4: Classification
Compute the likelihood ratio of a record pair and classify it as a match, non-match, or a possible match by using upper and lower cutoff limits.
Step 5: Evaluation
Evaluate result accuracy by computing sensitivity, specificity, and positive & negative predictive value.
Match
Non-match
Possible match
Review
Fig: A general approach to record linkage process

Fig. 1. A general approach to record linkage, consisting of five main steps: Preprocessing, Blocking/Indexing, Field Comparisons, Classification, and Evaluation.

B. Blocking/Indexing

Goal: Reduce computational complexity by grouping only similar records together before comparison.
Methods: Hash-based blocking (e.g., Soundex keys), sorted neighborhood approaches, or other indexing schemes.
Relevance to HW1: Especially critical when dealing with large datasets. A good blocking strategy can drastically speed up deduplication while maintaining acceptable recall.

C. Field Comparisons

Goal: Select which attributes or fields to compare (e.g., `NAME`, `TYPE`) and choose suitable similarity functions.
Methods: Character-based metrics (Levenshtein, Jaro-Winkler), token-based metrics (TF-IDF, cosine similarity), and phonetic encodings (Soundex, NYSIIS).
Relevance to HW1: Each name variant in the *test* data must be compared to potential matches in the *primary* and *alternate* files; careful selection of similarity metrics can catch subtle variations.

D. Classification

Goal: Decide which record pairs represent actual matches, non-matches, or uncertain matches requiring further review.
Methods:

- *Threshold-based* (e.g., Fellegi-Sunter upper and lower cutoff)
- *Supervised classification* (logistic regression, SVM, random forests)

Relevance to HW1: Once pairwise similarity scores are obtained, a threshold or classifier can be used to label pairs as matches (Y) or non-matches (N), with an optional "possible match" tier for manual review.

E. Evaluation

Goal: Assess the performance of the linkage system and tune parameters if necessary.
Methods:

- **Precision, Recall, F1-score** (preferable in class-imbalanced scenarios)
- **Sensitivity, Specificity**, or other domain-specific metrics
- **Clerical review**: Manually check borderline cases

Relevance to HW1: You will measure how well your deduplication and mapping tasks perform using metrics such as Precision, Recall, and F1-score. Simple accuracy can be misleading if there are far more non-matches than matches.

III. Similarity Scoring Methods

A. Character-Based Metrics

Levenshtein Distance. A common edit-distance-based metric is Levenshtein distance, defined recursively. Let s_1 and s_2 be two strings:

$$d_{\text{lev}}(s_1, s_2) = \begin{cases} |s_2|, & \text{if } |s_1| = 0, \\ |s_1|, & \text{if } |s_2| = 0, \\ \min\Big\{ d_{\text{lev}}(\text{tail}(s_1), \text{tail}(s_2)) + c\big(\text{head}(s_1), \text{head}(s_2)\big), & \\ \quad d_{\text{lev}}(\text{tail}(s_1), s_2) + 1, & \\ \quad d_{\text{lev}}(s_1, \text{tail}(s_2)) + 1 \Big\}, & \text{otherwise.} \end{cases}$$

$$c(a, b) = \begin{cases} 0, & \text{if } a = b, \\ 1, & \text{otherwise.} \end{cases}$$

where

$$c(a, b) = \begin{cases} 0, & \text{if } a = b, \\ 1, & \text{otherwise.} \end{cases}$$

This distance can be converted into a similarity score by

$$\text{Sim}(s_1, s_2) = 1 - \frac{d_{\text{lev}}(s_1, s_2)}{\max(|s_1|, |s_2|)}.$$

Jaro-Winkler Similarity. The Jaro similarity between two strings s_1 and s_2 is:

$$\text{Jaro}(s_1, s_2) = \begin{cases} 0, & \text{if } m = 0, \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right), & \text{otherwise,} \end{cases}$$

where m is the number of matching characters, and t is half the number of transpositions. The Jaro-Winkler similarity adds a prefix adjustment:

$$\text{JW}(s_1, s_2) = \text{Jaro}(s_1, s_2) + p \cdot \ell \cdot \big(1 - \text{Jaro}(s_1, s_2)\big),$$

where ℓ is the length of the common prefix (up to 4) and p is a constant (commonly 0.1).

B. Token-Based and Phonetic Approaches

Cosine Similarity. When comparing token-based vectors (e.g., TF-IDF), a common formula is:

$$\text{CosineSim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|},$$

where $\mathbf{a}$ and $\mathbf{b}$ are vector representations (e.g., counts or TF-IDF values) of the text.
Phonetic Encoding. Techniques like Soundex or NYSIIS convert words to phonetic codes before comparing. For instance, Soundex maps letters to digits, ignoring vowels (with some exceptions), to capture similar-sounding names.

C. Privacy-Preserving and Deep Learning Methods

Privacy-preserving record linkage uses techniques like Bloom filters to allow encrypted comparisons [7]. Recently, deep embedding models such as Sentence-BERT capture semantic similarity in structured records [8]. These can be especially powerful when dealing with more complex text attributes or cross-domain data.

IV. Match Decision Strategies

A. Threshold-Based Approaches

A widely used strategy classifies record pairs using one or more similarity thresholds. The Fellegi-Sunter model introduced the concept of upper and lower thresholds plus a "clerical review" region [9]. Practical systems often choose a single tunable threshold to balance precision and recall [10].

B. Supervised Classification Models

Supervised methods use labeled training examples to learn decision boundaries [11], [12]. Common classifiers include logistic regression, SVM, or random forests. These models produce match probabilities that can be thresholded for specific precision or recall targets.

V. Evaluation Metrics

Record linkage tasks often face a severe imbalance between matched and unmatched pairs, making simple accuracy misleading. Preferred metrics include:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

and

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Precision-recall curves and domain-specific constraints help select thresholds [6], [13]. Although ROC-AUC is popular, it can overestimate performance when positive pairs are relatively rare [13].

For the **HW1 tasks** (deduplicating the *primary* and *alternate* files, and mapping *test* data), collecting true matches vs. false matches allows computing TP, FP, FN, and TN, then deriving the above metrics.

VI. Conclusion

Record linkage requires careful orchestration across **five main steps**—preprocessing, blocking, field comparisons, classification, and evaluation—to handle real-world data challenges such as typographical errors and missing attributes. Threshold-based and supervised methods each offer trade-offs in complexity and interpretability, while metrics like precision, recall, and F1-score ensure a meaningful performance assessment. Emerging methods like semantic embeddings and privacy-preserving linkage show promise for future applications.

For the **HW1 tasks** of deduplicating the *primary* and *alternate* name files and mapping *test* data to original records, practitioners should:

1) **Preprocess** (normalize names, manage format inconsistencies),
2) **Block** similar candidate records for efficient comparisons,
3) **Compare** field values (using character-based and possibly phonetic metrics),
4) **Classify** pairs with threshold or supervised methods,
5) **Evaluate** using precision, recall, F1-score, and consider manual review if necessary.

This structured approach not only ensures robust linkage but also highlights areas for iterative improvement.

References

[1] M. A. Hernández and S. J. Stolfo, "Real-world data is dirty: data cleansing and the merge/purge problem," *Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 9–37, 1998.

[2] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.

[3] W. E. Winkler, "Overview of record linkage and current research directions," U.S. Census Bureau, Tech. Rep., 2006.

[4] T. N. Herzog, F. J. Scheuren, and W. E. Winkler, *Data Quality and Record Linkage Techniques*. Springer, 2007.

[5] W. E. Winkler, "String comparator metrics and enhanced decision rules in the Fellegi–Sunter model of record linkage," U.S. Census Bureau, Tech. Rep., 1990.

[6] P. Christen, *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012.

[7] V. Schnell, T. Bachteler, and J. Reiher, "Privacy-preserving record linkage using Bloom filters," *BMC Medical Informatics and Decision Making*, vol. 9, no. 1, p. 41, 2009.

[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[9] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[10] T. Panse, J. Papenbrock, and F. Naumann, "Evaluation of duplicate detection algorithms," Hasso-Plattner-Institut, Tech. Rep., 2021.

[11] R. Abramitzky, L. P. Boustan, and K. Eriksson, "To marry or not to marry: Matchmaking in the age of big data," *Historical Methods*, vol. 51, no. 4, pp. 203–217, 2018.

[12] Education Policy Initiative, "Pre-processing and linking: Overview of the MEDC matching process," University of Michigan, Tech. Rep., 2020.

[13] J. M. Hand and P. Christen, "A note on using the F-measure for evaluating data linkage algorithms," presented at the Newton Institute Workshop on Data Linkage, Cambridge, UK, 2016.