Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

Entity Resolution Pipeline: Five Stages, Worked Example

> Reliable Context is a kernel of truth: a system of record for organizations that fuse private and public information and must defend every value they act on. One canonical record per real thing, a cited source and an observation date behind every value, unknowns declared rather than left blank, and any past state of the record reproducible.

Published by Reliable Context · 2026-10-29 · entity resolution · entity resolution techniques · record linkage steps · entity resolution pipeline · deduplication record matching explained

Original article: https://reliablecontext.com/articles/entity-resolution-pipeline-standardize-block-compare-cluster



In this article

1. One real thing, many records
2. The worked example: six supplier records
3. Stage 1: Standardize
4. Stage 2: Block
5. Stage 3: Compare
6. Stage 4: Classify
7. Stage 5: Cluster, and why transitivity bites
8. Evaluation without fooling yourself
9. Human review and declared ambiguity
10. Where relationships replace merges
11. Checklist

One real thing, many records

Papadakis and colleagues define entity resolution as "the task of identifying different entity profiles that describe the same real-world object", and distinguish deduplication (matches within one collection) from record linkage (matches across two duplicate-free collections) (Papadakis et al., §1-2). In practice it runs as a pipeline of five stages:

```
standardize -> block -> compare -> classify -> cluster
```

Each stage fails in its own way, and each failure is invisible to the stages after it: a pair missed by blocking is never scored, and a clustering step that chains pairwise links can merge two legal entities that no single comparison would have merged.

The literature usually draws the pipeline with four stages. Binette and Steorts describe it as "attribute alignment → blocking → record linkage → canonicalization", where canonicalization is merging resolved records "to produce a single representative record" (Binette and Steorts, §2). The five-stage view splits their record-linkage stage into comparison, classification and clustering, because each fails differently. Canonicalization comes after and is out of scope here.

Stage	What it does	Typical failure	Where to audit it
1. Standardize	Parse and normalize fields	Over- or under-normalization changes what "equal" means	Spot checks of raw vs normalized values
2. Block	Choose which pairs to compare	True matches never become candidates	Blocking recall audit
3. Compare	Turn each pair into a comparison vector	Similarity levels that do not separate matches from coincidences	Level-by-level m and u review
4. Classify	Decide match, non-match or possible match	Thresholds set without regard to the cost of each error	Human review routing

Stage	What it does	Typical failure	Where to audit it
5. Cluster	Turn pairwise decisions into entities	Transitive chaining merges distinct entities	Cluster-level evaluation; merge or relate

The worked example: six supplier records

Everything in this example is fictional. Halden Therapeutics AG and Halden Pharma AG are invented Basel companies; the addresses, Swiss business identification numbers (UIDs) and source systems are invented for illustration.

A procurement team holds six supplier records from three systems:

ID	Name	Street	Postcode, city	UID	Source
S-001	Halden Therapeutics AG	Hafenstrasse 12	4057 Basel	CHE-111.222.333	ERP
S-002	HALDEN THERAPEUTICS A.G.	Hafenstr. 12	4057 BASEL	CHE111222333	Supplier portal
S-003	Halden AG	Hafenstrasse 12	4057 Basel	(empty)	AP invoice
S-004	Halden Pharma AG	Hafenstrasse 12	4057 Basel	CHE-444.555.666	ERP
S-005	Halden Therapeutics Inc.	200 Harbor Street	02139 Cambridge MA	(none)	ERP
S-006	Halden Therapeutics AG	c/o Treuhand Muster, Bahnhofstrasse 1	8001 Zürich	CHE-111.222.333	Supplier portal

The ground truth, as far as it is known: S-001, S-002 and S-006 are one company (S-006 carries a fiduciary's c/o address). S-004 is a different legal entity with its own UID that happens to share the building. S-005 is a separate US company in the same group. S-003 is an invoice record whose true identity is not established from the data shown.

Stage 1: Standardize

Standardization parses records into comparable attributes in canonical form. Binette and Steorts call this attribute alignment: free text is broken into "a common set of attributes" before any linkage method can run (Binette and Steorts, §2).

For the example, the rules are: lower-case and strip punctuation; split the legal form (AG, A.G., Inc.) into its own field; expand the street abbreviation "str." to "strasse"; reduce the UID to its letters and digits.

ID	Core name	Legal form	Street (normalized)	Postcode	UID (normalized)
S-001	halden therapeutics	ag	hafenstrasse 12	4057	CHE111222333

ID	Core name	Legal form	Street (normalized)	Postcode	UID (normalized)
S-002	halden therapeutics	ag	hafenstrasse 12	4057	CHE111222333
S-003	halden	ag	hafenstrasse 12	4057	null
S-004	halden pharma	ag	hafenstrasse 12	4057	CHE444555666
S-005	halden therapeutics	inc	200 harbor street	02139	null
S-006	halden therapeutics	ag	bahnhofstrasse 1	8001	CHE111222333

What goes wrong. Standardization decides what "equal" means for every later stage.

- *Under-normalization.* Without the UID rule, S-002's `CHE111222333` and S-001's `CHE-111.222.333` disagree as strings, and the strongest evidence in the dataset is lost.
- *Over-normalization.* A tempting rule removes "generic" words such as Therapeutics, Pharma or Holding. Applied here, S-001 and S-004 both become `halden ag` at the same address, and two companies with different UIDs look identical on name and address.
- *Lost information.* Dropping the legal form erases the difference between an AG and an Inc.; keep it as its own field.

A practical check: for each rule, count how many distinct identifiers each normalized value absorbs. A normalized name that absorbs several UIDs is a warning sign.

Stage 2: Block

Papadakis et al. state that without candidate selection "ER suffers from a quadratic time complexity, $O(n^2)$, as every entity profile has to be compared with all others", and that for deduplication the brute-force number of comparisons is $|E|\cdot(|E|-1)/2$ (Papadakis et al., §1-2). Splink's documentation gives the scale: 1 million input records "would generate around 500 billion pairwise record comparisons" (Splink, blocking rules). Blocking compares only records that share a key; records outside a common block "are automatically determined to be non-matches" (§2).

Rule 1 in the example: the first five letters of the core name plus the postcode.

Key	Records	Candidate pairs
`halde-4057`	S-001, S-002, S-003, S-004	6
`halde-02139`	S-005	0
`halde-8001`	S-006	0

Six records give 15 possible pairs; Rule 1 keeps 6. Papadakis et al. measure a blocking scheme with three ratios: pair completeness (PC, the share of true duplicate pairs that end up in some block, i.e. blocking recall), pairs quality (PQ, the share of candidate pairs that are true duplicates) and reduction ratio (RR, one minus

candidate pairs over all pairs) (§2). Counting only the three known duplicate pairs (S-001/S-002, S-001/S-006, S-002/S-006) and leaving S-003 aside because its truth is unknown:

- RR = 1 − 6/15 = 0.60
- PC = 1/3: only S-001/S-002 shares a block. The c/o address puts S-006 in another block, so its two true pairs are never compared.

Rule 2 adds exact equality on the normalized UID. Splink handles multiple rules the same way: it "will then produce all record comparisons that satisfy at least one of your blocking rules" (Splink, blocking rules). Rule 2 contributes S-001/S-006 and S-002/S-006, giving 8 candidate pairs, PC = 3/3 and RR = 1 − 8/15 ≈ 0.47. It works only because Stage 1 normalized the UID.

What goes wrong. Blocking errors are the hardest to see: a pair never compared produces no score and no review task. Papadakis et al. note that Standard Blocking's "effectiveness is very sensitive to noise, as the slightest difference in the blocking keys of duplicates places them in different blocks", and that schema-aware methods are "quite parameter-sensitive" (§3.2). Measuring pair completeness against adjudicated pairs is the subject of the blocking recall audit.

Stage 3: Compare

Each candidate pair becomes a comparison vector, one agreement level per attribute. Binette and Steorts write this as γ = (γ1, γ2, ..., γk), "where each γi corresponds to a comparison between a particular attribute" (§4.2). Splink builds each comparison from discrete comparison levels (for example exact match, Jaro-Winkler similarity above 0.95, all other) and notes that "the categories are discrete rather than continuous for performance reasons" (Splink, comparisons).

The example uses three comparisons:

Comparison	Levels
Core name	exact; partial (first token equal, rest differs); other
Street + postcode	exact; other
UID	equal; differ; missing on either side

The eight candidate pairs produce these vectors:

Pair	Name	Address	UID
S-001/S-002	exact	exact	equal
S-001/S-003, S-002/S-003	partial	exact	missing
S-003/S-004	partial	exact	missing
S-001/S-004, S-002/S-004	partial	exact	differ
S-001/S-006, S-002/S-006	exact	other	equal

What goes wrong. A level is useful only if it occurs at different rates among matches and non-matches. Levels that are too coarse blur that difference, and treating a missing UID as a disagreement would penalize every AP-feed record that never carries one.

Stage 4: Classify

Classification turns a comparison vector into a decision. Three families exist, and they complement one another.

Deterministic rules. Binette and Steorts observe that "in practice, the most commonly used record linkage methods are based on a series of deterministic rules involving the comparison of record attributes" (§3). A rule such as "same normalized UID means same company" is transparent, testable and, here, correct.

Probabilistic (Fellegi-Sunter). Each comparison level has an *m* probability, the probability of the observation given that the records match, and a *u* probability, the probability of the observation given that they do not. Splink describes *m* as "largely a measure of data quality" and *u* as "a measure of coincidence" (Splink, Fellegi-Sunter). The match weight is

$$M = \log_2(\lambda / (1 - \lambda)) + \log_2(m / u)$$

where λ is the prior probability that two random records match. Under the model's key assumption that comparisons are independent, the per-comparison weights add, and the probability of a match is $2^M / (1 + 2^M)$ (Splink, Fellegi-Sunter). In the original framework, a pair is linked above an upper threshold, declared a possible link between the thresholds and not linked below the lower one (Binette and Steorts, §4.2).

The *m*, *u* and λ values below are **illustrative**, chosen for the example, not estimated from data:

Level	m	u	log2(m/u)
Name exact	0.85	0.002	+8.73
Name partial	0.10	0.01	+3.32
Address exact	0.80	0.001	+9.64
Address other	0.20	0.999	−2.32
UID equal	0.95	0.0001	+13.21
UID differ	0.01	0.9998	−6.64
UID missing	(treated as neutral)		0.00

With λ = 0.001 the prior weight is log2(0.001/0.999) = −9.96. Summing:

Pair	Prior	Name	Address	UID	Total M	P(match)
S-001/S-002	−9.96	+8.73	+9.64	+13.21	21.62	>0.9999
S-001/S-006, S-002/S-006	−9.96	+8.73	−2.32	+13.21	9.66	0.999

Pair	Prior	Name	Address	UID	Total M	P(match)
S-001/S-003, S-002/S-003	−9.96	+3.32	+9.64	0.00	3.00	0.889
S-003/S-004	−9.96	+3.32	+9.64	0.00	3.00	0.889
S-001/S-004, S-002/S-004	−9.96	+3.32	+9.64	−6.64	−3.64	0.074

With an upper threshold of 8 and a lower threshold of 0, the three true duplicate pairs link, the two pairs involving Halden Pharma AG's UID are non-links, and the three pairs involving S-003 fall in the possible-link band.

The address weight shows the model's weak point: affiliated companies share buildings, so address agreement and partial name agreement are correlated among non-matches, yet the model adds them as independent evidence. Binette and Steorts, citing Smith and Newcombe (1975), note that the conditional independence assumption "may not hold in practice", and that Fellegi-Sunter relies on "crucial simplifying assumptions which are not expected to hold in practice" (§4.2).

Learned classifiers. dedupe learns field weights with "regularized logistic regression" from labelled pairs, and uses active learning to choose which pairs a person should label (dedupe, Matching Records). Ditto goes further: it fine-tunes pre-trained Transformer language models and casts matching "as a sequence-pair classification problem", serializing each record as `[COL] attr [VAL] value` tokens (Li et al., §2). The authors report improvements over the previous state of the art "by up to 29% of F1 score on benchmark datasets" and an F1 of 96.5% on matching two company datasets of 789K and 412K records. Ditto still needs a blocker: the paper's end-to-end system is "a blocker and a matcher". Binette and Steorts caution that the training data needed for deep neural networks "is rarely available for ER tasks" and that simpler classifiers are often preferred (§5.3).

Family	Needs	Strength	Typical failure
Rules	Domain knowledge	Explainable, stable	Brittle on noisy fields
Fellegi-Sunter	m, u, λ (estimated or labelled)	Calibrated weights, no labels required for estimation	Correlated fields, misspecified levels
Supervised / language model	Labelled pairs	Handles messy text	Label cost, drift, opaque errors

A defensible order is rules first, probabilistic weights second, learned matchers third, each later family scoring only the pairs the earlier one leaves undecided.

Stage 5: Cluster, and why transitivity bites

Identity is transitive (if A is B and B is C, A is C); pairwise scores are not. Binette and Steorts note that pairwise methods treat pairs "as being independent of one another, without accounting for the consequences of transitivity", and that clustering is commonly used "as a second step to probabilistic record linkage to enforce

transitivity of the output" (§6). dedupe's documentation gives an illustration: with A and B at 60% and B and C at 60%, the implied A-C probability under ideal conditions is 36%, and "a more principled answer does not exist because the estimated pairwise probabilities are not transitive" (dedupe, Grouping Duplicates).

The example produces exactly this triangle. Take A = S-001 (Halden Therapeutics AG), B = S-003 (Halden AG) and C = S-004 (Halden Pharma AG):

- A~B: M = 3.00, P = 0.889
- B~C: M = 3.00, P = 0.889
- A≁C: M = −3.64, P = 0.074

Suppose the pipeline uses a single threshold of P > 0.5 (M > 0) and then takes connected components of the link graph, one of the earliest clustering approaches (Binette and Steorts, §6.1). The six links with M > 0 form one component: {S-001, S-002, S-003, S-004, S-006}. Halden Therapeutics AG and Halden Pharma AG become one supplier, even though the model scored the direct pair at 7%, and their UIDs differ. The ambiguous invoice record is the bridge.

Remedies:

1. **Cannot-link constraints.** Two different validated UIDs forbid a merge regardless of path, so S-003 can join at most one side.
2. **Stricter cluster criteria.** Hierarchical clustering with a cut threshold (dedupe uses centroid linkage) or correlation clustering, which "maximizes the number of links within clusters plus the number of non-links across clusters" and is NP-hard so approximations are used, resist chaining (Binette and Steorts, §6.1; dedupe).
3. **Route the bridge to review.** S-003 is a possible link on two sides that cannot both be right; a person with more evidence (the invoice's bank account, the purchase order) should decide.

The correct output from the data shown is three resolved entities ({S-001, S-002, S-006}, {S-004}, {S-005}) and one record, S-003, explicitly marked unresolved.

Evaluation without fooling yourself

Pairwise precision and recall understate the damage above. In a large cluster, one wrongly merged record accounts for a small share of all pairs, yet a false merge like S-004's corrupts an entity that procurement, sanctions screening and spend analytics all depend on. Cluster-level metrics, error categories such as false merges and splits, and label sampling are covered in entity resolution evaluation beyond pairwise F1.

Evaluate each stage separately: a classifier evaluated only on candidate pairs cannot see what blocking lost.

Human review and declared ambiguity

The possible-link band exists because some pairs should not be decided automatically. In Binette and Steorts' account, the goal of Fellegi and Sunter's procedure is "to minimize the number of possible links" while controlling both error rates (§4.2). Which possible links deserve a reviewer, and when "unresolved" is the right

final answer, are the subject of entity resolution human review. The key design choice is that "not yet known" is a recorded state, as for S-003, rather than a default merge or a silent split.

Where relationships replace merges

Halden Pharma AG and Halden Therapeutics Inc. are not duplicates of Halden Therapeutics AG, but they are not unrelated either. A pipeline that knows only "same" and "different" will force them into one or the other. The better output is a typed relationship (affiliate, subsidiary, co-located) between distinct entities. The decision framework for organization records is in merge or relate.

Checklist

1. Normalization rules are checked on samples, and identifiers are normalized before use as blocking keys.
2. Blocking uses several OR-combined rules, and pair completeness is measured against adjudicated pairs.
3. Comparisons have an explicit "missing" level, and m and u are estimated, not guessed.
4. Correlated fields (name and address of affiliates) are checked.
5. Thresholds produce three outcomes, and possible links go to review.
6. Clustering enforces cannot-link constraints and does not rely on connected components alone.
7. Evaluation includes cluster-level metrics, not only pairwise F1.
8. Unresolved records are declared unresolved; related entities are linked, not merged.

The output this pipeline aims for matches what Reliable Context's published guarantees describe: one canonical record per real thing, with unknowns declared, never blank.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://arxiv.org/abs/1905.06167
2. https://arxiv.org/abs/2008.04443
3. https://moj-analytical-services.github.io/splink/topic_guides/blocking/blocking_rules.html
4. https://moj-analytical-services.github.io/splink/topic_guides/comparisons/comparisons_and_comparison_levels.html
5. https://moj-analytical-services.github.io/splink/topic_guides/theory/fellegi_sunter.html
6. https://docs.dedupe.io/en/latest/how-it-works/Matching-records.html
7. https://arxiv.org/abs/2004.00584
8. https://docs.dedupe.io/en/latest/how-it-works/Grouping-duplicates.html

THE PUBLISHER

About Reliable Context

Reliable Context is a system of record for organizations that fuse private and public information and must defend every value they act on.

The guarantees

One canonical record per real thing: duplicates resolve by rule, ambiguity is shown rather than hidden, and related organizations relate without ever being merged. A verbatim, cited source and an observation date sit behind every value. Unknowns are declared as not found, blocked or not applicable with a reason, never left as an empty cell. Any past state of the record is reproducible, together with what changed since.

Where it runs

It runs inside the customer's perimeter, one database per organization. Permitted sources, model routing, reader clearance and retention are set as the customer's own policies, and every value is exportable with its chain of custody for an auditor or regulator.

What it is not

It is not a vector database or a RAG framework, not a data catalog, MDM or data-quality suite, not a CRM or data-enrichment vendor, and not a chatbot or AI assistant. Where a reader needs one of those, the honest answer is to say so and point them there.

Visit

See Reliable Context.

Reliable Context is in production use in two sectors; more than six thousand of its identity decisions were made by rule and none has been reversed. Articles published here explain general practice, cite their primary and academic sources, and name the jurisdiction of any regulatory rule they describe. They are not legal, regulatory or compliance advice for a specific organization, and requirements change.

Website: https://reliablecontext.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.