Profisee Group, Inc.

BEYOND MATCHING

ENTERPRISE GOLDEN RECORD MANAGEMENT

Beginning with examples of contrasting data scenarios, this paper illustrates that data duplication problems vary greatly. Those examples are then used to show that the accuracy of a solution must be measured according to the problem it is solving. For the complex data problem, technical approaches to fuzzy matching algorithms are next explored and demystified. With this, the application of fuzzy matching techniques is described as used to match and de-duplicate records. And finally, this paper shows that the solution for data duplication extends beyond matching. Worrying about one technology (such as the matching algorithm), combining multiple piecemeal solutions, or misapplying solutions leads to suboptimal results at a high cost. This paper contends that a complete solution and maximum return on investment is provided only by a Golden Record Management solution that is integrated with a master data management platform.

WRITTEN BY VAL LOVICZ
and edited by Emily Byars

§Who Needs Master Data Management?

Duplicate master data and missed opportunities for revenue, margin, and efficiency have been linked in case study after case study.

Business environments change rapidly from mergers and acquisitions, creating numerous database applications when coupled with a proliferation of specialized line-of-business systems. The result is a commonplace struggle for many large businesses today: organizations face increasing inefficiencies and missed opportunities from fragmented data living in multiple applications. In a traditional architecture environment, data is not integrated so that a customer is uniquely identified and consistently described across all applications, or that a product is tracked in uniform throughout the company. An organization that lacks such data cohesion cannot possibly recognize the total value of their business.

Countless industry experts have analyzed enterprise data duplication problems and their proposed solutions. As a result, many vendors now market matching solutions to solve the data duplication problem. These solutions, their benefits, their drawbacks, and their inner-workings are the subject of this paper. To learn about how data duplication issues affect enterprises in your industry, visit www.Profisee.com.

Beginning with examples of contrasting data scenarios, this paper illustrates that data duplication problems vary greatly. Those examples are then used to show that the stability of a solution must be measured according to the problem it is solving. For the complex data problem, technical approaches to fuzzy matching algorithms are next explored and demystified. With this, the application of fuzzy matching techniques is described as it is used to match and de-duplicate records. And finally, this paper shows that the solution for data duplication extends beyond matching. Worrying about one technology (such as the matching algorithm), combining multiple piecemeal solutions, or misapplying solutions leads to suboptimal results at a high cost. This paper contends that a complete solution and maximum return on investment is provided only by a Golden Record Management solution that is integrated with a master data management platform.

§ What Is Enterprise Golden Record Management?

Golden Record Management (GRM) is a Master Data Management (MDM) approach that includes,

- A central repository of linked records with a representative master or "golden record" for each linked group
- Transparency in results, recording which records are linked, how they were linked, and by whom
- Publishing of relevant information to all subscribing systems, and publishing of master hierarchies to BI applications and the data warehouse
- Mastering processes to create the best representative golden record for each unique entity
- Harmonization processes that selectively update source records in MDM via rule logic as a signal to propagate changes back to source systems
- Multi-user, secure workflows for reviewing proposed matches, approving or rejecting matches, and creating new linkages by exception
- Additional data quality and MDM processes, such as address standardization, to augment and improve the shared golden records

Each of these capabilities is integral to an enterprise-grade GRM solution, no matter the name or price. In the first sections of this paper, the two most common data scenarios are closely examined to illustrate key distinguishers of an enterprise level data problem, for which a GRM solution is imperative.

§ The Data Duplication Problem

Most enterprises are already painfully aware of the costs, errors and missed opportunities associated with duplicate data. Records for customers, suppliers, products, and more are duplicated in multiple systems due to a proliferation of operational systems and mergers and acquisitions. Often no mechanism exists to uniquely identify each entity across systems and no proactive steps are taken to prevent the creation of duplicate records.

Examples of resulting business process problems from data duplication are endless:

- A regional construction materials company allows a customer to open a new line of credit at one division while the customer is already on credit hold at another division.

- An insurance company does not have a "lookup before create" check in place and therefore creates a duplicate member record. When this client files a claim, the company inadvertently processes their single claim twice.

- A global cosmetics company is unable to provide aggregated product and customer reporting, as each region records entities in a different way, with a different name, in their own systems. 20,000 real products are reported as 900,000 products worldwide. 2,000 customers are reported as 30,000 unique customer records.

- A healthcare provider cannot assess their billing liability based on insurance coverage because it lacks a process which ensures that only unique or master patient records are calculated.

- A large global music provider cannot combine its complex music sourcing information (tracks, artists, labels, legal entities) into a global media list across all channels.

- An investment bank with a heavy mergers and acquisitions growth model is unable to efficiently calculate total exposure to a given individual or organization across the entire company.

- A large hardware and software reseller that competes primarily on product price must track more than 50 providers with multiple SKUs for thousands of products. Managing their data manually in Excel with add-ons, Access DBs, and macros causes lost price margins and lost business.

- An international manufacturing firm focused on growth through acquisition spends nine months per acquired company to integrate customer and product masters into their environment. With up to 10 acquisitions per year, duplicate data chokes IT resources and consumes the data quality budget.

Having identified their data duplication problem, many organizations embark on solutions with varying success. Most solutions are sold with record matching technologies and processes, but the "data quality solutions" category has ballooned to include a vast range of products. As a result, many matching solutions are fit to clean only a narrow range of data types. But, not all matching problems are equal.

§Matching Problem Definition

Before discussing the strengths and weaknesses of matching and de-duplication solution approaches, it is helpful to understand the differences among matching scenarios.

Not All Matching Problems Are Equal

Consider the following two scenarios:

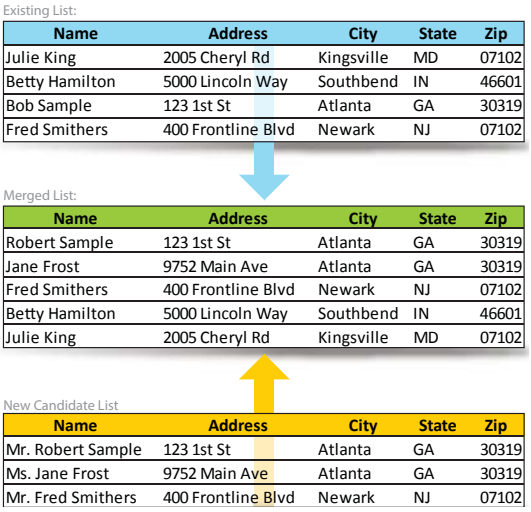
- **Scenario 1 [*Isolated*]**: A data steward must compile distinct mailing lists sourced from multiple third-parties into a single, trustworthy list for a static direct-mail campaign.

- **Scenario 2 [*Enterprise*]**: Multiple data stewards must collaborate to link and unify customer records across several ERP instances and systems for ongoing, dynamic use.

These scenarios highlight strongly different goals and objectives.

The Isolated Scenario

In the *Isolated* scenario, there is a single stakeholder (a marketing person generating new leads). The stakeholder has the simple goal of merging the lists to avoid duplicate mailings and reduce mailing costs. If her de-duplication process accidentally over-matches or under-matches records, she simply misses one opportunity for mailing (or executes a duplicate mailing). The stakeholder can be solely responsible for the output data and the consequences for errors are low. She does not care about linking the original records to the output records, and, at her discretion, she may discard duplicate source records. Once she creates the output merged list, she continues to use the resulting list for campaigns and embarks on a new one-off batch merging process the next time a new mailing list is purchased.



Existing List:

Name	Address	City	State	Zip
Julie King	2005 Cheryl Rd	Kingsville	MD	07102
Betty Hamilton	5000 Lincoln Way	Southbend	IN	46601
Bob Sample	123 1st St	Atlanta	GA	30319
Fred Smithers	400 Frontline Blvd	Newark	NJ	07102

Merged List:

Name	Address	City	State	Zip
Robert Sample	123 1st St	Atlanta	GA	30319
Jane Frost	9752 Main Ave	Atlanta	GA	30319
Fred Smithers	400 Frontline Blvd	Newark	NJ	07102
Betty Hamilton	5000 Lincoln Way	Southbend	IN	46601
Julie King	2005 Cheryl Rd	Kingsville	MD	07102

New Candidate List

Name	Address	City	State	Zip
Mr. Robert Sample	123 1st St	Atlanta	GA	30319
Ms. Jane Frost	9752 Main Ave	Atlanta	GA	30319
Mr. Fred Smithers	400 Frontline Blvd	Newark	NJ	07102

Figure 1 - Isolated Matching Scenario - data example

The Enterprise Scenario

The *Enterprise* scenario requires de-duplication of records from multiple source systems. In this case, merging and discarding records is not an option. The ERP systems already have transactions (e.g., sales orders) against the customer IDs within those systems and for that reason, those source records cannot be purged or have their IDs changed. This scenario requires a mapping or linking approach to de-duplication where each source system ID (and record) is retained and linked to a new global customer identity. Furthermore, the *Enterprise* scenario has multiple stakeholders across the business that are impacted by the use of the customer data. Multiple departments such as sales, customer service, and manufacturing all fulfill customer obligations based on the ERP customer records, and they all contribute to creating and maintaining those records. The stakeholders must be involved and take ownership in the linking process, and results must be transparent.

Source Records:

ERP1:

CUST.CUST_NAME	CUST.CUST_ID	AM.ADDR1	AM.CITY	AM.STATE_PROV	AM.POSTCODE	AM.COUNTRY	AM.PHONE1
Bike World	1014	60025 Bollinger Canyon Road	San Ramon	CA	94583	US	264-555-0150
Bike World, LLC UK	2002	123123 Workingham Rd	Reading	Berkshire	RG61WG	UK	(+44) 0844 800 2400
Bike World	2201	123123 Workingham Rd	Reading		RG61WG	UK	(+44) 0844 800 2400

CRM1:

Customer.CustName	Customer.ID	Addr.AddressLine1	Addr.City	Addr.State	Addr.PostalCode	Addr.Country	Contact.MainPhone
Bike World	2001	60025 Bollinger Canyon Road	San Ramon	CA	94583	US	264-555-0150
Bike World, Reading	2005	Workingham	Reading			UK	(+44) 08448002400
B.W.LLC UK	2154	123123 Workingham Rd	Reading		RG61WG	UK	
World of Bikes LLC	2008	123 Totenam Road	Reading	Berkshire	RG61WG	UK	

Record Source	Name	Code	Address Line 1	City	State or Prov	Postal Code	Country	Main Phone	Master	Score	Match Status
MDM	**Bike World**	**M1014**	**60025 Bollinger Canyon Road**	**San Ramon**	**CA**	**94583**	**US**	**264-555-0150**			**Master**
ERP1	Bike World	1014	60025 Bollinger Canyon Road	San Ramon	CA	94583	US	264-555-0150	M1014 {Bike World}	1.00	Auto-Approved
CRM1	Bike World	2001	60025 Bollinger Canyon Rd	San Ramon	CA	94583	US	264-555-0150	M1014 {Bike World}	0.94	Auto-Approved
MDM	**Bike World, LLC UK**	**M2002**	**123123 Wokingham Rd**	**Reading**	**Berkshire**	**RG6 1WG**	**UK**	**(+44) 0844 800 2400**			**Master**
ERP1	Bike World, LLC UK	2002	123123 Wokingham Rd	Reading	Berkshire	RG6 1WG	UK	(+44) 0844 800 2400	M2002 {Bike World,	1.00	Auto-Approved
ERP1	Bike World	2201	123123 Wokingham Rd	Reading		RG6 1WG	UK	(+44) 0844 800 2400	M2002 {Bike World,	0.99	Auto-Approved
CRM1	Bike World, Reading	2005	Wokingham	Reading			UK	(+44) 0844 800 2400	M2002 {Bike World,	0.85	*Proposed*
CRM1	B.W.LLC	2154	123123 Wokingham Rd	Reading		RG6 1WG	UK	(+44) 0844 800 2400	M2002 {Bike World,	0.85	Approved
CRM1	**World of Bikes LLC**	**2008**	**123 Toteham Road**	**Reading**	**Berkshire**	**RG6 1WG**	**UK**				**Unique**

Figure 2 - Enterprise Matching Scenario - data example

Not All Matching Solutions Are Equal

Choosing the most effective matching solution for your data problem is a question of ROI. How much is your data problem costing your enterprise and how much time and money does implementing the matching solution cost? When choosing a solution, consider these and the following factors.

Process and Usage

What is the process and how will the results be used? The *Isolated* and *Enterprise* scenarios differ greatly in the way results are processed and consumed.

For instance, in the *Isolated* scenario, the stakeholder may simply feed the results to a bulk mailing house and no other contributors are involved in the process.

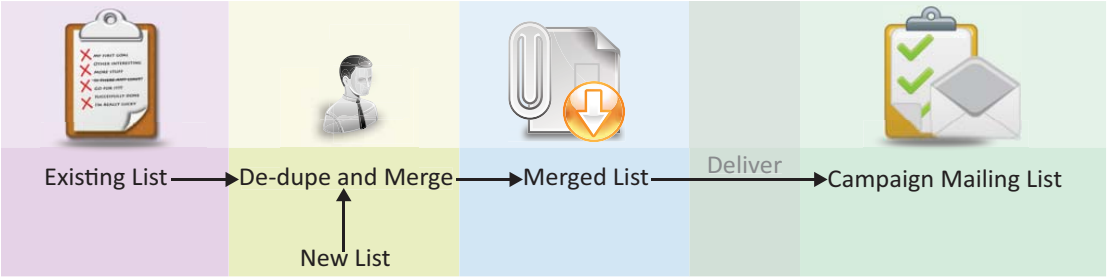


Figure 3 - Isolated Matching Scenario - process example

In the *Enterprise* scenario, many users and systems must consume the information, driving the need for a "unified view". Operational systems may need real-time feeds to align changes to data such as credit rating across linked customer records. Also, clear distinctions must be made where data should be different due to different purposes in local systems. The data warehouse needs a hierarchy consolidation point which aggregates the linked customer IDs from multiple systems.

Because business users own the data, it is imperative that they have full visibility and participation in the matching process. The business users' acceptance of automated matches, their approval of any questionable matches, and their processing of any manual matching assignments is critical; bad results from black-box data quality tools will inevitably fall on IT's head. Figure 4 shows an MDM-oriented architecture that is ideal for an environment in which many users and applications contribute and consume shared data.

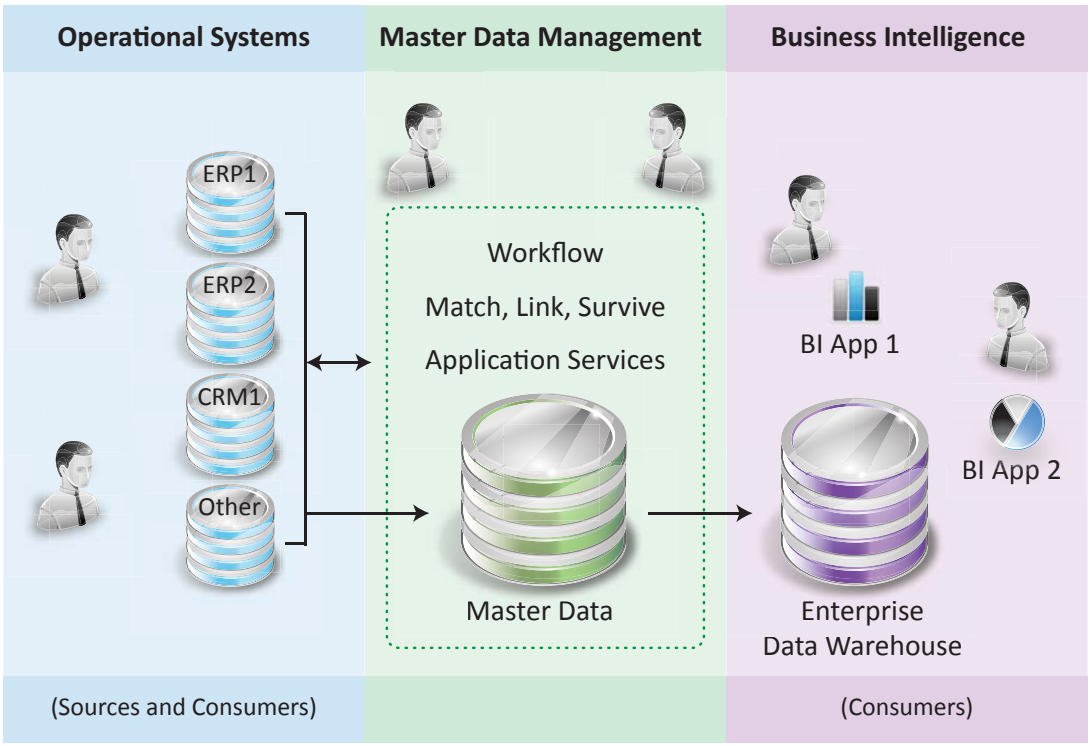


Figure 4 - Enterprise Matching Scenario – process and architecture example

Frequency of Change

The *Isolated* scenario involves an infrequent purchase of a bulk mailing list. In this scenario the data doesn't change much at all. This scenario is perfectly suited to full, batch processing.

Conversely in the *Enterprise* scenario, the ERP records are likely changing and growing every day. It's important in this scenario to incorporate incremental updates and additions and to support proactive "lookup before create" requests in real time to prevent duplicates when possible.

Contrasting Solution Approaches

Clearly, the *Isolated* scenario, with its limited requirements and data volumes, should be solved using simple desktop matching and data quality tools. Implementing anything more complex would significantly drive down ROI. The *Enterprise* scenario, however, requires not only a scalable and reliable matching process, but a vast array of other complex capabilities. A powerful matching algorithm is one of many important parts that make up a complete MDM solution. Enterprise data quality problems require a GRM solution – a holistic MDM approach.

§Demystifying Fuzzy Matching Algorithms

In the following section, solutions to the data duplication problem are examined by first reviewing the technical approaches to fuzzy matching and how they contribute to a record matching solution. Building on this review, Golden Record Management is then explored and additional requirements and features of a complete Enterprise GRM solution are investigated.

Matching Accuracy Terms and Concepts

Before diving into the weeds of matching algorithms, this section reviews matching accuracy concepts and related terms, which are important to the technical discussion of matching approaches.

When deciding if any two records represent the same thing, there are two choices:

- The records match
- The records do not match

Each of these choices introduces two possible errors:

- *False positive* (a.k.a. over-match) – records were matched, but do not represent the same thing
- *False negative* (a.k.a. under-match) – records were not matched, but do represent the same thing

Generally, altering an automated matching process to reduce one type of error increases the other type of error. However, the total number of errors may be reduced by,

- Improving the data to be matched (improving the matching "signal-to-noise ratio")
- Gathering the uncertain records for human review and decision making ("matching workflow")

Business objectives often drive whether false positives or negatives are the better compromise. If a mailing list is under-matched, as in the *Isolated* scenario, a few customers may receive duplicate mailings and some extra postage savings are missed. The consequences and cost of an under-match are low.

Similarly, the relative cost of errors determines how much expense is desired to improve the signal-to-noise ratio. Matching two different customer records in the *Enterprise*

scenario may result in improper calculation of sales discounts, applying a credit hold to the wrong customer, and other expensive missteps. In this case, spending extra resources to improve the matching signal-to-noise ratio and implementing a matching workflow will have a worthwhile return on investment.

Technical Approaches to Matching

Matching engines commonly employ fuzzy matching algorithms, scoring similarity on a scale basis. This section contains a technical examination of some standard fuzzy matching algorithms and highlights the most successful approaches to employing these in the record matching process.

Similarity

Fuzzy matching algorithms based on edit distance or set intersections compute the similarity between two compared strings. These algorithms have been around for decades. They helped drive innovation and accuracy in Web search engines, spell checkers and data mining tools. A few of these algorithms have even been around since the 19th century and were created for statistical uses completely unrelated to string comparison.

1. *Edit distance*

Edit distance is a way of calculating the difference between two strings by counting the smallest number of single character changes, deletions, or insertions required to transform one string into the other.

Examples of edit distance algorithms include:

- Hamming distance
- Levenshtein distance
- Damerau-Levenshtein distance
- Jaro distance
- Jaro-Winkler distance

Hamming, Levenshtein and Damerau-Levenshtein are all closely related. Each results in a count of single character edits. A lower count shows greater similarity, while a higher count shows lower similarity.

As an example, the Levenshtein distance between two strings: "Paul" and "Pawly" is 2.

1. Count one change to substitute "w" for "u"

2. Count one change to add "y".

The Levenshtein algorithm is not normalized based on relative string length, so it could and should be extended to account for the relative length of strings.

In contrast, Jaro and Jaro-Winkler distance algorithms also count the number of changes between two strings, but these algorithms use an equation to turn the result into a normalized score that ranges from 0 (no similarity) to 1 (exact match). Jaro-Winkler is a variation of Jaro that places more emphasis on the beginning of the string. For comparing edit distance calculations, Jaro-Winkler is considered better suited for short strings such as person names.[1]

Understanding how edit distance algorithms apply to field records requires a number of basic considerations:

- The base algorithms do not account well for multi-word fields on their own. Some additional logic is needed such that "John Smith" and Smith, John" are considered identical, or nearly identical.

- The algorithms must analyze more potential edits as the string size grows, so calculation times increase exponentially as the compared strings get larger and more dissimilar.

- Comparing every possible pair of strings in a long record list is not practical. For example, to compare every item to every other in a list of 100 would require up to 4950 comparisons (sum of the sequence 99, 98, 97, … 1).

$$f(X) = \sum_{n=1}^{X-1} n = \frac{n(n-1)}{2}$$

Edit distance calculations are well suited to finding candidates in a spell-check operation because of their word-oriented nature and their ability to account for small typos. Spell-check benefits from the fact that (1) thing is compared to (N) possibilities, whereas as matching compares [Nx(N-1)/2] records. In both cases, the value of (N) must be reduced, if possible, to make the comparison perform. If you have ever typed an incorrect word for which spell-check couldn't suggest an obvious correction, then you can appreciate the downside of reducing the comparison set.

2. Token-based distance

As mentioned above, text fields with multi-word strings pose additional challenges for edit distance calculations. When comparing "John Smith" with "Smyth, John", it may be more important to pair up the similar words and ignore the order of words. Breaking up strings into multiple words and comparing on a word-by-word basis is called *token-based* matching. To ensure that "Smith" is compared to "Smyth" and "John" is compared to "John" may require a *recursive matching scheme,* as described by Monge and Elkan.[2] This requires a first pass to find the best pairings of words among the two strings and then a second pass to aggregate the similarity scores of those best matches. The drawback is the quadratic time complexity as the string lengths increase.[2]

3. Set intersection

Set intersection similarity is a statistical technique that has many applications, but can also be applied to the comparison of strings. Example calculations include:

- Jaccard index or Jaccard similarity coefficient
- Tanimoto distance, or Tanimoto similarity ratio
- Sørensen similarity index

These calculations all work in almost the same manner. Multiple subtle variations of these calculations have been defined.

Paul Jaccard was a professor of botany and developed the Jaccard index to compute similarity of two data sets, such as the similarity of species within communities.[1] To define the Jaccard index in simple terms, it is the ratio of the intersecting members of two sets to the union of all members of two sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

To apply this index, a set of observable features must be defined for the subjects being compared. Each set either contains the feature or not. For example, when comparing species in two communities,

- Community A contains species {A, B, C, F, G}
- Community B contains species {B, C, D, E, G, H, I}
- There are 3 intersecting species {B, C, G}

- There are 9 unique species among the groups {A, B, C, D, E, F, G, H, I}

Therefore, the Jaccard index for similarity of Community A to B is 3/9, or 0.333.

To apply this technique to text strings, species are replaced with words or pieces of words and communities are replaced with text strings. Compare the following example text.

1. the sly fox

2. slyish fox, the

In a simple case, we could compare the presence of whole words only.

Observed Values	In Text 1	In Text 2	Common to Both
the	X	X	X
sly	X		
slyish		X	
fox	X	X	X

The similarity index is then 2/4 = 0.5.

For more granular results, the Jaccard algorithm can be used to look for more than just words. In the next example, words are broken into substrings or grams, resulting in bigger sets for comparison; a minimum string size of 3 was considered, while overlapping and partial strings were ignored for simplicity.

Observed Values	In Text 1	In Text 2	Common to Both
the	X	X	X
sly	X	X	X
ish		X	
fox	X	X	X

The similarity index is then 3/4 = 0.75.

Clearly, the method with which a string is broken into meaningful parts is important and drives the reliability of set comparisons on strings. The concept above could be extended to every possible substring of a certain size. Generally, including more granular parts of strings within the comparison samples drives better results. However, increasing the number of pieces sampled from the string can lead to a slower matching process and a proliferation of set data.

Set intersection calculations are well suited to string matching, given their versatility, applicability to multi-word strings, and normalization of scores based on string sizes. Furthermore, these calculations are more deterministic than edit distance. This kind of comparison also lends itself to pre-conversion of all string data into numerical vectors (one-dimensional arrays), making it well-suited to software algorithms and storage. Therefore, set intersection techniques are one of the best methods of matching a population of strings.

Phonetics

Phonetic algorithms have also been employed in fuzzy matching solutions; they encode or transform words for comparison based on pronunciation.[2, 3] Some of the most common algorithms for English are,

- Soundex
- Metaphone and Double Metaphone
- New York State Identification and Intelligence System (NYSIIS)

Phonetic algorithms generally work by substituting characters (or numbers, for certain character occurrences) such that two similarly pronounced words are represented by the same value. For example, the name Clare and Clair are two alternatives for the same name with the same pronunciation. Phonetic algorithms attempt to transform these words into identical encoded strings that yield an exact match. The Soundex code C460 represents both Clare and Clair.

To illustrate how these algorithms work, here are the basic rules for American Soundex encoding as summarized by Wikipedia.org.[4] Note that rule 3 is not in perfect sequential process order.

1. Retain the first letter of the name and drop all other occurrences of [a, e, i, o, u, y, h, w].
2. Replace consonants with digits as follows (after the first letter):
    a. b, f, p, v => 1
    b. c, g, j, k, q, s, x, z => 2
    c. d, t => 3
    d. l => 4
    e. m, n => 5
    f. r => 6
3. Two adjacent letters (in the original name) with the same number are coded as a single number; also two letters with the same number separated by 'h' or 'w' are coded as a single number, whereas such letters separated by a vowel are coded twice. This rule also applies to the first letter.
4. Continue until you have one letter and three numbers. If you run out of letters, fill in 0s until there are three numbers.

From the prior example of Clare, the encoding proceeds as follows:

1. Clare => Clr
2. Clr => C46
3. {no change}
4. C46 => C460

Phonetic algorithms are generally developed with a specific language in mind, such as English. Therefore, applying the rules to words from other languages can produce unreliable results.[2] With so many countries having significant immigration, few lists of person names are immune to multi-cultural spellings and pronunciations.

While these algorithms can be helpful to an overall solution, over-reliance on phonetics can be very limiting, as these algorithms do not handle common data variation problems like transpositions, typos, nicknames, and so forth.

Phonetic substitutions add value as a part of an overall matching strategy. For example, there is no drawback to having an additional surname Soundex code within the data set to support another form of comparison. However, due to phonetic algorithms' limitations, similarity should be used as the primary fuzzy matching technology. The inclusion of phonetics is helpful but not necessary to achieve complete and accurate results.

More Than Just Data Quality

In a real-world scenario, data sets contain multiple field data records and these basic fuzzy matching concepts must be expanded and combined to handle such complex data. Strategies for such an expansion are diverse and many miss the mark. But no matter how capable a matching solution or data quality tool is, it cannot be effective without business user visibility and participation, direct interaction with data, formal mastering and harmonization processes, and a transparent and secure workflow for manually analyzing the matching process. Additionally, without supplementary services like robust address verification, the matching process can be unsuccessful by default with bad information.

§Applying Fuzzy Matching to Data Records

While many matching solutions employ the same core matching calculations, how they overcome the surrounding challenges varies greatly. Some solutions break records up into manageable pieces to achieve smaller processing batches, while others achieve a manageable number of candidate records to be compared in the fuzzy stages of matching by relying on pre-selection techniques. The following section examines some of these common strategies with their benefits and compromises.

String Similarity Calculations

Many record matching solutions use string similarity calculations as a foundation. Some of the challenges when such a strategy is applied to record matching are,

- Handling records involving multiple fields of various types and inconsistent population
- Removing noise or errors from fields that lead to matching errors
- Scaling the matching computation when it could lead to an explosion of "match index" data, which takes a significant and exponentially increasing time to calculate and space to store

Scoring and Thresholds

Normalized similarity algorithms compute a similarity score classifying the relative similarity of two strings. The scores generally range from 0% to 100% or 0 to 1.0, where 0 represents complete dissimilarity and 100% (or 1.0) represents an exact match. Similarity scores are then aggregated or averaged across multiple tokens within fields and across multiple fields. The scores are also weighted or adjusted based on the relative importance of certain terms or fields.

A matching threshold is chosen so that a score below the threshold is treated as a non-match and a score above the threshold is treated as a match.

As mentioned previously in the *Matching Accuracy Terms and Concepts* section, raising or lowering a single threshold simply trades one type of error for another (e.g., increasing over-matches to reduce under-matches). To counteract this trade-off, a second threshold may be introduced. This second threshold is referred to as the "auto-approval threshold", as match links above this threshold are considered automatically approved while the "soft matches" falling between the match threshold and the auto-approval threshold need business user review for approval or rejection.

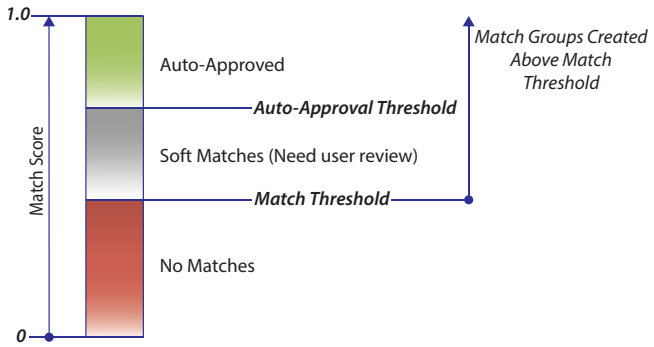


Figure 5 - Applying multiple matching thresholds

Some matching solutions pursue error avoidance by focusing on the matching signal-to-noise ratio. Relying too heavily on the automated matching engine leads to overlooked opportunities for soft matches, the human workflow element. In an enterprise context, user approval of questionable matches and user mapping of manual matches are as imperative as the matching algorithm. For example, a business selling to both Skype and Microsoft may want to link their customer records for Skype and Microsoft based on the knowledge of a merger. Therefore, a matching solution that does not provide a workflow experience for human decisions is inadequate and drives excessive add-on application development to facilitate this requirement.

When using a three-tiered / two-threshold approach, the lower threshold is set just low enough that under-matches are avoided and the auto-approval threshold is set high enough such that over-matches do not get auto-approved. This leaves the soft match category containing matches that are either,

- Okay, but require user confirmation
- Over-matched and must be broken and moved

A complete matching solution comes with a matching workflow process that enables these kinds of decisions out-of-the-box. Workflow cannot be an afterthought left for

custom development and implementation. A secure, intuitive, multi-user application to review, approve and reject matching results is essential to your project's success.

Multiple Fields and Field Types

Similarity based on a single field rarely produces desirable fuzzy match results. When matching records using multiple fields, not all attributes are equally valuable, but should be weighted based on the specific business requirements and data. For this reason, a record matching solution should be fully capable of handling multiple attributes of varying types.

Identity Versus Non-Identity

One general way to classify matching fields is *identity* versus *non-identity*.

Identity fields are those that signify (or nearly signify) a unique record. Clearly a social security number or customer ID can be a perfect unique key for People; these are the best form of identity fields. The sole reason for using a fuzzy match on these key fields is to account for errors (i.e., typos) in the source data. Other fields, such as a person's full name or street address, are also considered identity fields because they come relatively close to being unique across a population. Though two people can have the same name and 123 Main Street can exist in multiple cities, the number of over-matches on these fields will be relatively low when compared to a field such as city of residence. Because these fields are more unique, they can also have a higher matching threshold applied. Therefore, identity fields such as these should be called out and leveraged to find the best match group candidates across an entire population.

Non-identity fields are those that are almost always assumed to be non-unique across a population of records. These include fields such as city or zip code of residence, date of birth, gender, and so forth. Due to high likelihood of shared values, these fields are poor choices for forming match groups. However, they are great for un-forming them. For example, if two people share the same name and have a similar street address name, they may appear similar, but if they live in completely different cities and have different birth dates, then the desired conclusion is that these records are not the same person. Using non-identity fields for refinement, rather than initial candidate selection, is a best practice.

Based on the use of non-identity fields to dispute rather than form matches, the thresholds can be much lower. For example, a match similarity threshold on City may be set to 0.5 without fear of over-matches because it only compares records within groups formed by identity fields and seeks to confirm or deny those matches. Note that this technique is distinctly opposite of "blocking or pruning" methods that attempt to group records (e.g., by city, state, product family, etc.) before fuzzy matching.

Whole Match Versus Partial Match

It's important to distinguish fields containing meaningless key values (i.e., product category ID = "1003") from text values (i.e., product category description = "Mountain Bikes"). This contrast in scenarios highlights the need for whole matching versus partial matching. Whole and exact matching would be prudent in the category ID example so that you don't match Mountain Bikes (ID="1003") with Tube Socks (ID = "1030"). This example assumes that category IDs were not directly entered by users and do not suffer from typos or transposition errors.

Blank Values

As the number of fields included in the match is increased, the likelihood of encountering blank fields or fields populated with meaningless values (such as "unknown") is increased. The matching process should allow for ignoring of blank values in some fields so that the match decision falls to the available attributes.

Multiple Passes

When considering a real-world matching scenario involving many attributes, multiple overlapping passes may be desired. As an example, patient records are often matched exactly by social security number where available, followed by other matching processes that leverage name and address for records not including a social security number. Both matching passes in this example should contribute records into the same pool of match groups, each scoring the matches differently based on the available fields. Each matching process should filter the appropriate input records.

Combining Field Type and Multi-Pass Intelligence

Ideal matching processes allow multiple match fields to contribute appropriately based on unique business requirements and the nature of the data being matched.

The process should:

- Efficiently process identity and non-identity field groups in parallel
- Avoid blocking or pruning that may miss matches
- Support whole-versus-partial and exact-versus-fuzzy matching within the same process
- Smartly account for blank values

- Flexibly support multiple matching strategies contributing to the same matching result set

Noise Removal, or Improving the Signal-to-Noise Ratio

Just like any other process, improving the matching process input can improve the matching process output. In other words, reducing garbage in reduces garbage out. There are many ways to increase the "signal" (the real information) that indicates a match, while also reducing the "noise", or misleading information that obscures the real matching signals. In the following sections, strategies for such a goal are reviewed in detail.

Basic Normalization

In an enterprise matching engine, some basic refinements should happen automatically as data enters the model. This includes removal of punctuation or meaningless characters, as well as normalization of case. Clearly this step should not involve changes in the source data record; what's good for matching is not always better for data quality (or other uses of the data). Also, the normalization should have standards and defaults, but ultimately it must be configurable to accommodate the business scenario. For example, the "#" symbol may be meaningless noise in a customer name field, but could be very significant in a part number field.

Synonym Replacement

Another source data modification is standardization of terms. We refer to this as *synonym replacement*. For example, "street", "strt", "str" and "st" in a US address could all refer to the same equivalent string: "street". To both improve the matching signal and reduce noise, the equivalent terms should be both standardized and reduced to the smallest length. Reducing all street strings to "st" may seem counter-intuitive because it is contrary to normal data quality exercises. That's true, but reducing the string length makes it less significant in the matching process. When matching "Main St" to "Main Street", the "Street" term should not be considered more significant than the word "Main" simply because of its length. The street term should be considered, just not as much as its relative string length would indicate. This example also highlights why simply improving the quality of the source system fields and standardizing on longest terms like "Street" is not best for the goal of matching. Both endeavors add value, but they can often be separate and distinctly different endeavors.

Synonym Removal

To extend the synonym replacement concept even further, some terms are purely noise and should be removed altogether from the matching input stream. An example is the word "Corp" or "Corporation" in the context of a company name. The reason is that this word is often arbitrarily omitted. A customer record is just as likely to contain the name "Microsoft Corporation" as it is to contain "Microsoft". A person would intelligently recognize these names as the same company with high certainty, even if they were unfamiliar with the company name. Therefore, rather than standardizing Corporation to Corp. and Incorporated to Inc., these terms should be removed altogether to reduce noise.

Synonym Building

The process of synonym building can be automated by gathering a distinct set of terms and phrases from the data. In an identity field, terms that appear in at least twice as many records as would be expected in one distinct group are likely to be noise and should either be reduced in size or eliminated. In a non-identity field, repetition is expected (e.g. many people live in the same city). Therefore, the unusual terms should be reviewed for potential correction to a standard. For example, there may be 1000 occurrences of "Atlanta" and a couple occurrences of "Al<u>t</u>anta".

Probability

One way to reduce the flow of noise into the matching process is to use probability to discount frequent terms in the data set. If empirical evidence suggests, for example, that approximately 20 customers contain the word "services", then that single customer record has only a 5% chance of matching any other record with the same word, based on that word alone. Therefore, the word "services" is useful for hinting that a record may relate to the group of 20 customers but it does not serve well to positively identify the record group that it matches. To apply probability, the matching process can either disregard the word "services" or proportionately weight it to give it a lower relevance.

Inverse Frequency Weighting

As mentioned above, proportionately weighting terms based on their frequency in a population can reduce the noise of meaningless common terms in a data set. The Web search community is well aware of this phenomenon when searching for terms in documents and much has been published on the IDF weighting technique, or "inverse document frequency". The inverse document frequency is the log of the ratio of the total number of documents (D) to the number of documents (d_t) containing a term (t) within an entire corpus or population.[1]

$$IDF_t = \log\left(\frac{D}{d_t}\right)$$

The inverse document frequency weighting may be applied to terms in similarity scoring by comparing the total number of strings to the number of strings containing a term. For example, an infrequent term appearing in 5 strings out of 10,000 will have an IDF weight of log(10000/5) = 3.30 while a frequently encountered term appearing in 5000 strings out of 10,000 will have an IDF weight of log(10000/5000) = 0.30. Applying the IDF weighting within the overall similarity score reduces the noise of unimportant terms.

Sampling and Feedback Loop

Whether using synonyms, probabilities and/or inverse frequency weighting to adjust the signal, using the actual data and using knowledge of the data characteristics to change the inputs and improve the results is an ideal feedback loop.

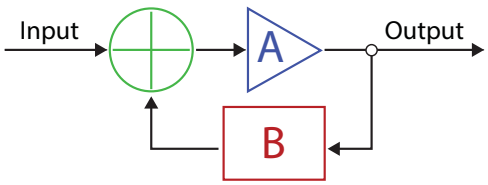


Figure 6 - Process feedback loop

Use of sampling and feedback in the form of synonyms or probability is a good way to lessen the impact of noise terms and phrases within the matching input to yield better quality output.

The techniques described above are often used in combination. For example, IDF can automatically adjust the frequency of common terms; however, without the standardization of terms (e.g. "svc" = "services") via synonym replacement, some terms will not be fully adjusted based on their true occurrence frequency. Furthermore, some terms should be disregarded based on knowledge of the data set and not frequency.

Importance of Multiple Fields

The above techniques are generally effective to improve the matching results from a single field only. The impact and need for these techniques is diminished as more fields are added to the matching process. For example, it doesn't matter if noise in the name field leads to the over-matching of two customer records when address fields are available to confirm or dispute the matching signal from the name field. Therefore, these techniques should be employed when there is little to match on other than a single field such as

name, but the importance is greatly diminished as soon as there are two or more input fields.

Introducing Verification and Standardization

Verifying data with third party services such as an address verification service to correct and/or standardize addresses is also valuable for improving the signal-to-noise ratio, but over-reliance on this capability alone can be costly and ineffective for matching.

Many solution vendors that once focused on this data quality activity now sell address verification as a solution for matching. Address verification services rely heavily on address cleanup and parsing to produce exact matches. These are expensive and ineffective solutions if taken in isolation.

Generally, address verification solutions make external lookup calls via Web services. This architecture is essential given the pace of change with new addresses, postal code changes, boundary changes, etc. Unfortunately, most solutions charge fees per record for these lookups. Matching solutions that rely on address verification can drive significant annual transaction fees because all records must be frequently rechecked along with new records. Suppose 100 Peachtree St., Atlanta, 30319 becomes 100 Peachtree St., Dunwoody, 30328 due to incorporation or boundary changes. Matches will be missed unless all historical records are re-verified periodically along with new records.

The capabilities of address verification services vary greatly. Some are better than others at fuzzy matching an input address to their database of good addresses and thus the fuzzy matching problem then relies on the capability of the provider's platform, so the ability to control over or under matches and engage users within workflows is diminished.

Some services can effectively handle the parsing of address information from single or inconsistently used fields while others cannot. Address inconsistency examples include,

- Full address in one field
- Address in generic "Line 1", "Line 2", and "Line 3" fields
- Street address in Line 2 or Line 3
- Missing city, state, or zip information

Therefore, it is important to understand the variability and relative quality of source addresses and select an address verification solution that can achieve acceptable results. A solution that blends the right amount of address verification to improve the matching signal-to-noise ratio without excessive transaction fees and add-on components is ideal.

§Adding Golden Record Management

The complete solution to data duplication and fragmentation extends beyond matching. Furthermore, point solutions to matching, data quality, address verification, and integration do not yield a rapid application solution, and they will drive significant implementation costs and recurring service fees. A complete solution is provided by a golden record management process and an MDM-based approach described here.

1. The canonical model

One of the great values of MDM is the canonical data model, defined in common business terms, where multiple source systems may be mapped and loaded. To speed early implementation, the MDM hub is non-invasive where necessary, loading and managing source records with or without modifications to existing source systems. The MDM hub may also provide seamless integration with record creation processes as source systems are ready to evolve and make proactive lookup-before-create calls. The canonical model and authoritative central hub of MDM are key enablers for SOA in modern integration architectures.

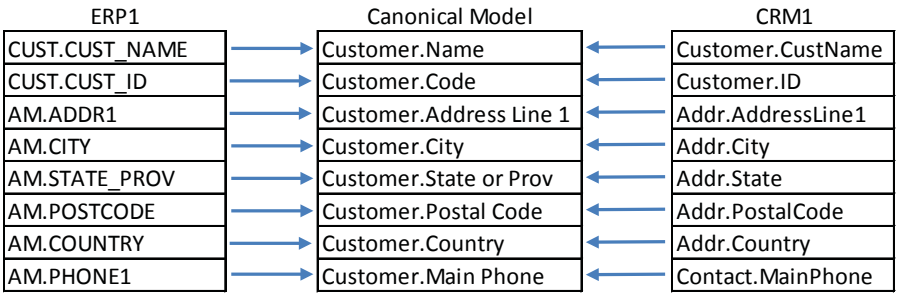


Figure 7 - Mapping systems to a canonical model

2. Handling results with MDM

Business users must see, understand, and participate in the inputs, decisions, and outputs of matching and other data quality processes for ownership and acceptance. For matching and standardization processes, this means the source records, changes, augmentations, links, scores, and statuses should be tracked and visible as master data.

(1) Match Attributes						Control Attributes			
(3) Customer.Name	Customer.Address Line 1	Customer.City	Customer. State or Prov	Customer. Postal Code	Customer. Country	Bill-to Location Master	Bill-to Location Match Score	Bill-to Location Match Status	Record Source
Bike World	60025 Bollinger Canyon Road	San Ramon	CA	94583	US	M1014 {Bike World} (4)	(5) 0.95	Auto-Approved (6)	ERP1 (7)
Bike World	60025 Bollinger Canyon Rd	San Ramon	CA	94583	US	M1014 {Bike World}	0.94	Auto-Approved	CRM1
Bike World, LLC UK	123123 Wokingham Rd	Reading	Berkshire	RG6 1WG	UK	M2002 {Bike World, LLC UK}	0.99	Auto-Approved	ERP1
Bike World	123123 Wokingham Rd	Reading		RG6 1WG	UK	M2002 {Bike World, LLC UK}	0.99	Auto-Approved	ERP1
Bike World, Reading	Wokingham	Reading			UK	M2002 {Bike World, LLC UK}	0.85	Proposed	CRM1
World of Bikes LLC	123 Toteham Road	Reading	Berkshire	RG6 1WG	UK	M2002 {Bike World, LLC UK}	0.75	Proposed	CRM1
B.W.LLC	123123 Wokingham Rd	Reading		RG6 1WG	UK	M2002 {Bike World, LLC UK}		User Mapped	CRM1

(1) Matching strategy constructed to match on Customer.Name and address fields
(2) Matching control attributes visible to business users and data stewards
(3) Model defined in common business terms
(4) New master records are constructed based on matching and survivorship criteria and thresholds; source records are linked to master records
(5) Match scores classify relative similiarity of source-to-master
(6) Match status (based on preset match threshold and auto-approval threshold) reveals the merge output of the matching strategy — high similarity records automatically get approved, soft matches are propsed as matches but require data steward attention
(7) Record source identifies the origin of the record, and is utilized in survivorship strategies to automatically inherit specific attributes (e.g., Customer.Name) from source systems designated as the best represntative of the attribute (e.g., ERP1)

Figure 8 – Visibility into "Bill-to Location" matching results

The MDM hub provides the first and immediately available repository of results and actionable dashboards of the results. Secondly, the data warehouse and business intelligence applications will rely on the record linkages in MDM to consolidate data into universal views of distinct customers, products, locations, etc. While the source transaction or fact data will be related to individual source system IDs, the BI hierarchies will leverage the linkage hierarchies provided by MDM's golden record relationships.

Matching Using Mapping and Linking

A mapping and linking approach is essential when integrating data from multiple enterprise systems. Each source system will need to retain its IDs for entities such as customers or products. The links from source records to golden records in MDM will serve as an auditable, functional way to relate and consolidate data from multiple systems into one unified view.

Record links provide the flexibility to break and change links over time. Processes which discard detailed data in favor of the consolidated result only do not provide this flexibility to change.

While record links may form simple two-level hierarchies such as source to master, they can also be complex, multi-level links where the output of one match becomes the input to another. For example, customer bill-to addresses may be consolidated into a unique customer address master, which in turn is consolidated into a distinct customer account representing that distinct customer, regardless of location.

The mapping and linking approach also supports the use of multiple matching strategies applied to subsets of records that yield a single set of results. For example, different matching criteria could be applied to different source records by comparing and matching them to a universal set of common master records (e.g. distinct customer addresses).

In other words, one set of mappings can be produced by multiple alternative matching passes.

Survivorship

Once matching produces linked records, it is important to merge records in the right way. Merging should be performed as a "logical merge" which preserves source records and an audit trail rather than a "physical merge" which discards or destroys source records and provides little visibility into the process.

"Survivorship" is a logical merging process that includes two distinct steps.

1. *Mastering*: creating the best possible representative master record from the source records contained in the match group

2. *Harmonization*: selectively updating source record attributes in a match group based on better / standardized data of the master record

Mastering to Create a Golden Record

Master records have the autonomy to be a composite record or a superset of information that is broader than any one source system. For example, a CRM may have the best support contact information and an ERP instance may have the best ship-to address, but only the MDM golden record may have both pieces of information merged from the appropriate source systems. Mastering rules determine which attributes should be mastered from what sources and who should win in the cases of multiple contributors or conflicts. The mastering rules must also have multi-field awareness such that whole addresses are selected from the same chosen source record. The MDM hub also provides an audit trail showing which sources contributed which pieces to the master record.

Harmonization for Update or Correction of Source Systems

With harmonization, authoritative information from a master record is selectively pushed to one or more linked source records. You may ask, "Why change a source system record directly in MDM?" This is an auditable way to trigger an update to the source system. An event handler looks for the update on the source record in MDM which sends a message to the source system to change the value (or sync with the MDM hub). This approach provides transparency because the recorded transactions in MDM show that the source value was changed by harmonization.

The combination of mastering and harmonization, under the survivorship umbrella, forms a closed-loop process to take data values from designated authoritative systems

and propagate those values to other systems that need them. For example, a central corporate ERP instance may hold the authoritative credit hold indicator for customer records which needs to be pushed out to other ERP instances to block new orders. A change in a customer's zip code may need to be distributed to all related address records across multiple billing systems.

Process Definition and Configuration

Defining matching and survivorship processes has much commonality across multiple businesses and domains. However, each business's data and data model is significantly different. Those business users who posses knowledge of the data should be involved in defining and adjusting these processes. Matching and survivorship should not be locked inside a black-boxed, IT-coded solution. Business users should be able to define the processes, make incremental test runs, view the results and quickly refine to reach the end result of well-matched records.

A secure user interface drives the definition of the matching process within the context of the data. Which columns and types will be matched? What attributes within the MDM data model will be used to show the results of linking, approval status, number of related records, and so forth? While there are best practices for matching strategies on data sets such as addresses, customers, and patients, the users have flexibility to quickly adjust common strategies to suit their data perfectly.

Matching strategy definitions and processes are aware of incremental runs and prior results. New records may be introduced daily, hourly or by the minute. The process should incorporate new data into prior matched records and preserve prior user approvals or overrides unless there is a business decision to start over with new definitions.

Business requirements may also drive multiple matching strategies on the same data set. These may be complementary or completely independent. For example, one strategy and set of result attributes could group distinct customers based on personal details while another strategy groups households consisting of one or more customers.

Business Rules for Extensibility

A general deterministic rules engine complements and extends the capabilities of matching processes if the rules may be applied before and after matching and easily triggered as part of a larger task. Some rules may create concatenations or parsed fields, which are helpful to matching. In other cases, rules could trigger special behavior after matching, such as downgrading auto-approved matches to proposed for user review based on certain record anomalies that deserve attention. In each case, applying the rules in the context of MDM provides transparency and enables matching processes and general business rule processes to work together on the same data set.

Workflow Processes for User Review and Ownership

One of the most important aspects of MDM and matching solutions is the human workflow element. A matching solution implementation that overlooks this deliverable or relies on custom application development and implementation could easily create a significant time and cost overrun and risk the entire project. The workflow process is one of the most expensive components that the solution must deliver.

No matter how good the matching process, users will always need to review and approve some of the matches. Additionally, users with knowledge of the data may need to create special manual links between records that matching rules could not create.

Users should not only manage the exceptions and questions in the matching process, they should be able to review and challenge any match links across the data set where they have ownership. Business users will not accept the results (and regulatory rules may even be violated) if the right users can't trace, accept, and own the results.

The bigger the data problem, the more users will be involved. Users may be segmented among multiple matching purposes as well as multiple data sets (e.g. someone owns the match links for customer records from her division). Therefore, the MDM stewardship user interface and workflow processes must be designed with many concurrent users in mind. The system must allow granular security by data sets, rows, and columns and must allow segmentation into specific roles (approvers versus requesters versus augmenters). The system requires built-in notifications and issue tracking to ensure that users are aware of data needing their attention.

Ultimately, the ongoing stewardship of the data and maintenance of matched sets must continue to be effective over time to achieve maximum ROI. This can only happen with a streamlined and easy process that the business users will accept and embrace.

§Conclusion

Matching is just the beginning of a complete solution to data duplication, albeit an important one, because one time, on-demand data cleansing is inadequate. Without an infrastructure for ongoing data management, your project's ROI diminishes drastically as data is duplicated once again in months, weeks, or even days. If a solution is to sufficiently manage enterprise master data, it must come with a multi-strategy matching engine as part of a larger solution. This matching engine must be capable of matching a monolithic body of tens of millions of data members out-of-the-box, lest your IT team be required to contribute significant manual coding to an already resource expensive project. An enterprise worthy solution must also integrate data quality services into the central hub, not offer them as an external add-on. Survivorship, mastering, and harmonization are imperative capabilities for creating golden records and for publishing your scrubbed data to downstream applications or upstream source systems. These tasks cannot be performed without a native, multi-user, secure workflow for the data cleansing process. And if the out-of-the-box workflow does not provide transparent results of linkage and change history

for the data, any bad matches will inevitably be blamed on IT. Finally, after all data is scrubbed, diverse applications must be able to easily consume it. An enterprise GRM application easily pushes relevant information to all subscribing systems and publishes master hierarchies to BI applications and the data warehouse.

If your data is to be consumed by multiple parties, demands frequent updates, or requires a workflow for secure editing, then only GRM can support your requirements. Though implementing such solutions can be laborious, a complex problem is only repaired by a conglomerate solution. Any data cleansing tool that relies too heavily on matching, data services, or any other single component is inadequate and will inevitably yield a poor return. In an enterprise environment, a Golden Record Management solution that is integrated with a master data management platform is the only truly complete solution for untrustworthy data.

§About Profisee

Profisee is a master data management software company focused on delivering enterprise-grade MDM capabilities through its Master Data Maestro software suite.

Master data consists of the information that is key to the core operation of a business. Master data may include data about people (customers, employees, vendors), places (branches, stores, cost/revenue centers), and things (products, contracts, accounts, assets). Master data also defines relationships between objects, such as reporting hierarchies, rollups or bill of materials, which provide the context of how data is derived and viewed within an organization.

Profisee was formed from the management team of Stratature, the leading MDM software provider acquired by Microsoft in 2007. Profisee is a Microsoft Gold ISV (independent software vendor) partner, a member of the exclusive Microsoft MDS ISV program, and was named "Cool Vendor in MDM" in 2011 by a leading industry analyst firm.

With a strong heritage in master data management, Profisee is uniquely positioned to help customers build a solid foundation and successful path to reach their master data management goals.

§References

1. *Adaptive Name Matching in Information Integration.* **Bilenko, Mikhail and Mooney, Raymond.** 5, s.l. : IEEE, 2003, IEEE Intelligent Systems, Vol. 18.
2. *The field matching problem: Algorithms and applications.* **Monge, Alvaro E. and Elkan, Charles P.** s.l. : AAAI, 1996.
3. **Wallwork, John Anthony.** *The distribution and diversity of soil fauna.* s.l. : Academic Press, 1976. pp. 44-46.
4. **Ross-Patterson, et.al.** Soundex. *Wikipedia.* [Online] 2004-2012. [Cited: March 29, 2012.] http://en.wikipedia.org/wiki/Soundex.
5. The Soundex Indexing System. *National Archives.* [Online] 2007. http://www.archives.gov/research/census/soundex.html.
6. **Garcia, Dr. E.** Term Vector Theory and Keyword Weights. *Mi Islita.* [Online] October 27, 2006. http://www.miislita.com/term-vector/term-vector-1.html.
7. *Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines.* **Jaccard, Paul.** Bulletin de la Société Vaudoise des Sciences Naturelles, 1901, Vol. 37.
8. **Coetzee, et al.** Levenshtein distance. *Wikipedia.* [Online] 2003-2012. [Cited: March 29, 2012.] http://en.wikipedia.org/wiki/Levenshtein_distance.