Deep Data Insight Master Person Index

Jeewa Perera, Indula Kulawardana and Eshan Mewantha Herath

January 19, 2018

Abstract

In current data driven business environment Master Person Index (MPI) applications are becoming more and more integral to all industries ranging from healthcare to education where information of the same person is recorded in different places without any reference linking these together. Deep Data Insight MPI system analyzes data sources, learns and recalls previously seen records, creates and maintains a master index and references to each original data source. In addition to these, the system is also equipped with end user assistance through text based search and facial recognition allowing the end user to query the system using full/partial information of the person or using images of the person. This system provides fast, effective and secure MPI solutions for any industry including Healthcare, Insurance, Supply Chain and Education. In these industries, this system has the impact in increasing the efficiency and effectiveness of person/entity coordination, management, analytics, etc. The customizable nature of the system provides the flexibility of making efficient integrations with existing as well as new systems.

1 Introduction

1.1 Master Person Index

In practical environment, it is common for one person to be recorded multiple times in the same organization/system. For example, for a healthcare provider which has multiple healthcare facilities throughout the state or the country, it is common for a patient to be registered multiple times depending on factors such as patient providing slightly different information intentionally or by mistake, clerical errors, communication errors, etc. This causes issues such as loss of patient treatment history, erroneous information generated for managerial tasks, etc. The task of a master person index (MPI) is to be a single point of reference for all the data sources. A MPI solution maintains a repository of person records which contains a single record per person with a unique identification number, relevant personal details and the references of each of these records to the original data sources' records. In addition to creating and maintaining a singular point of reference, a MPI system is expected to be able to retrieve person's records when an end user (e.g., a clerk at registration desk) queries. As there may be situations where the query will be having different information than what's in MPI, due to reasons such as use of a nick name, misspelling, use of initials, different address and/or date formatting, etc., the system is expected to be capable of finding the correct record from the MPI even when the available query information is incomplete and/or contains errors. The MPI system, when retrieving a person record, as a practice provides a confidence score (also referred to as a matching score), which provides the end user a measure of how confident the system is on the matching of the input information and the record returned.

1.2 Significance of Deep Data Insight MPI system

This is capable of creation and maintenance of unique records using multiple data sources, learning and updating its own internal representations when new records are added, retrieval of one or more closely matching records when a query is given with fully/partially complete information.

In addition to conventional text based search, this system has the additional capabilities of searching the MPI for unique records based on facial recognition. This reduces delays and errors which occur at the time of entering query. (e.g., The clerk might mishear the information the person is providing due to the noisy environment in patient waiting area.). With facial recognition integrated, the patient can just look at the camera on clerk desk, and the system captures an image of his/her face, run it through the recognition process and retrieve the relevant patient record.

This solution is configurable to work with either file based or database table based input/output combinations. The fields on which the matching is to be done and what type of matching approach is to be taken for each data field is also configurable thus providing it the flexibility of being applied to any type of a dataset. These, together with effective design aspects has enabled it to be a scalable, fast to implement, cost effective, customizable solution which is vendor-agnostic, completely inter-operable with other systems through its standard API.

Key aspects of MPI system :

- Customizable for any industry (Healthcare, Insurance, Banking, Engineering and Education) based on the requirements.
- Provides a fast, accurate and effective solution.
- Ensures security and privacy of data.
- Facilitates cleaning up duplicate records and eliminating duplicate registrations.
- Allows user intervention on ambiguity resolution.
- Enables text based searching as well as query through facial recognition.
- Reduces operational costs and errors occurred due to duplicate person records.
- Provides unique person identifier for fast, effective and accurate data operations.
- Uses cutting edge technologies

1.3 Overview of the system MPI Architecture

Primarily, this solution's MPI creation and maintenance part (other part being MPI query through text or facial recognition) is designed to function on four main phases, namely, configuration phase, ambiguity resolution phase, training phase and matching phase. The main processes of Internal Record Linkage is spread across the above four phases in order to make the functionality more user friendly as well as efficient. Figure 1 illustrates a high level architecture of the MPI solution.

High level descriptions of each of the aforementioned phases are shown below which are followed by the description of the system based on record linkage stages.

Configuration phase : At the beginning of system set up, this phase is handling all the customizations related to types of inputs/outputs, fields to be used in matching process and type of matching algorithm for each field, upper and lower thresholds of confidence scores, etc. Unless the tables or the data structures change, this is often a onetime task.

Ambiguity Resolution phase : During the initial run, this phase is not met since there aren't any records yet in the system. After the first Matching phase, the system obtains records in its Master Person Index and also references to each of the records in MPI from one or more records in input data sources. For example, in the previous healthcare provider scenario, two records from two healthcare facilities may be of the same patient who have visited both facilities over the years, thus both of these will be referencing to a unique record in MPI. And each reference also carries the confidence score which is a value between 0 and 100, where 100 means a complete match and 0 means a complete non-match. Based on the upper and lower confidence scores (set at the configuration phase) the system can decide on its own whether an incoming record is a new one or a

match for an existing one. During the matching process, records which are having a score between upper and lower threshold are flagged by the system for manual inspection. Ambiguity resolution is the phase where an employee would go through the flagged records and resolve the ambiguity using his/her experience and domain knowledge. This phase plays a partial role in Record Pair Classification stage of record linkage process while the rest of that stage is covered by Matching phase. Ambiguity Resolution phase takes place after each Matching phase and is immediately be followed by a Training phase before next Matching phase begins.

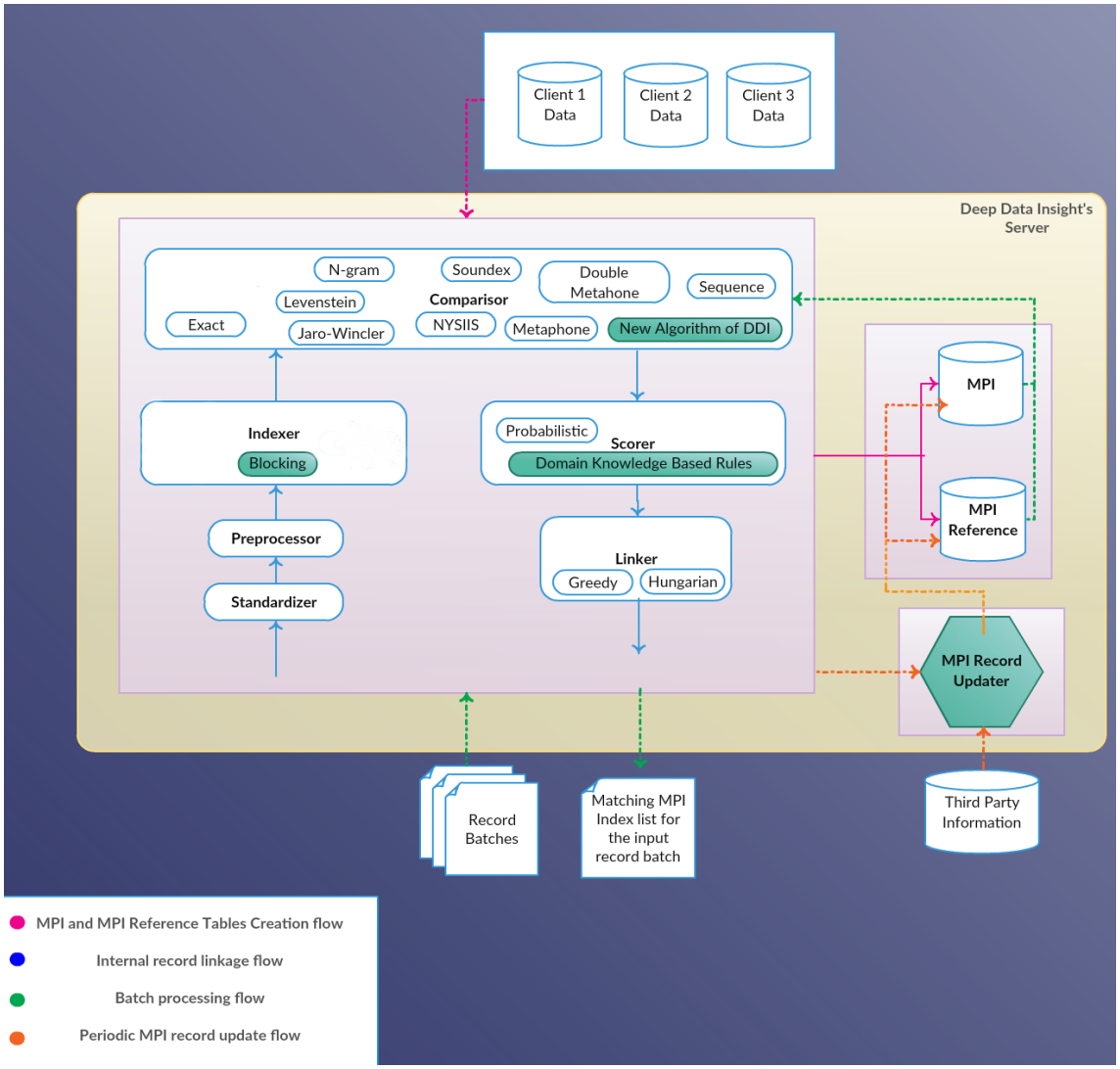

Figure 1: Deep Data Insight MPI System Architecture

Training phase : After ambiguity resolution phase, the module is trained with all the entries in the MPI. The objective of this phase is as follows. During the matching phase (and also during query time), the system cannot match and compare the input record with each and every record of the MPI to find the match. This is time consuming thus causing the system to be less efficient. Here, the system applies machine learning, specifically the unsupervised learning approaches in clustering, to learn by itself a effective representation of the MPI records. This covers a major part of the indexing stage of record linkage process, where common approach is to index using rule based blocking, this system utilizes the power of both rule based and machine learning approaches. Also, the system generates all the pre-calculations and matching criteria required for the next matching phase. The use of machine learning and pre-calculations reduce the number of records to which the input has to be matched as well as the number of calculations needed to be done per match, thus increasing the efficiency of the system.

Matching phase : The matching process is executed on an input data set, where the output is the same data set updated with the identification number of the record which it is matching the most and the respective confidence score. This is the phase which contains majority of the record linkage process, covering stages such as preprocessing, record pair generation, field comparison, record pair comparison and record pair classification. All of these stages are described in the following section.

2 Overview of the system in terms of Record Linkage stages

A proper implementation of a MPI system needs to cover the stages of record linkage process. As mentioned earlier, for the effectiveness and user friendliness, this MPI solution operates in phases which are different from standard record linkage stages. This is due to the fact that the system is designed by combining certain stages of the record linkage flow together to provide it more flexibility. Following section describes the system in terms of record linkage stages to highlight which phases of it are handling each record linkage stage.

1. Data Pre-processing

   During the configuration phase, the system is configured to conduct matching on certain fields; and therefore all the inputs needs to be preprocessed to validate and/or transformed to the aforementioned format. The Data pre-processing stage carries out these tasks and accepts or rejects input based on certain rules. This stage is handled by the Matching phase of the system.

2. Indexing

   Indexing is the second stage of Record Linkage process. This MPI solution applies Blocking as the indexing method. Simply, Blocking can be described as the process of generating sub datasets from a dataset in a methodical manner. As an example (a rule based blocking), if first letter of the surname is used for blocking, whole dataset is partitioned into 26 sub datasets according to the letters of the alphabetical. The training phase of the system is responsible for this index stage.

   Main purpose of blocking is to reduce the time taken when matching a new record, by reducing the number of records it has to be matched with. As mentioned earlier, the system is utilizing a combination of both rule based and machine learning based approaches to increase the accuracy of the blocks generated. A purely rule based approach usually generates blocks which end up still having a large number of entries thus causing a considerable delay in matching and also in certain cases may miss the closest records entirely. To solve this issue this solution is equipped with configurable blocking criteria where the rules can be configured based on the type and nature of the data that the system is handling. The MPI record set is first sliced using these rules and each slice of the data set is sent through a machine learning algorithm which sort them into different more fine tuned blocks within each data slice.

   When it comes to the use of machine learning in creating blocks, the system consists of the options of activating either k-means clustering or hierarchical clustering while the number of clusters and the fields to be used in clustering algorithm are also configurable. Thus giving it the flexibility of adjusting according to the data set type and size during the configuration phase.

   In fact what is achieved through indexing (Blocking in this case) is spatial partitioning of the MPI record set. Spatial Partitioning is used when a single data point (a record) can be viewed in a multidimensional space. After identifying blocking variables (blocking rules), these variables are used as inputs to a Spatial Partitioning process. Main objective of this partition process in blocking stage is to generate partitions in this space such that similar records are placed in the same partition. Mainly, there are two partition methods, Hard

partitioning Method and Fuzzy partitioning Method. In hard partition method, each record is placed in only one partition and each partition is assigned a number. In fuzzy partitioning Method, each record is placed in multiple partitions and each partition boundary is assigned a number. Following Figure 2 and Figure 3 illustrates examples of hard and fuzzy partitioning respectively.

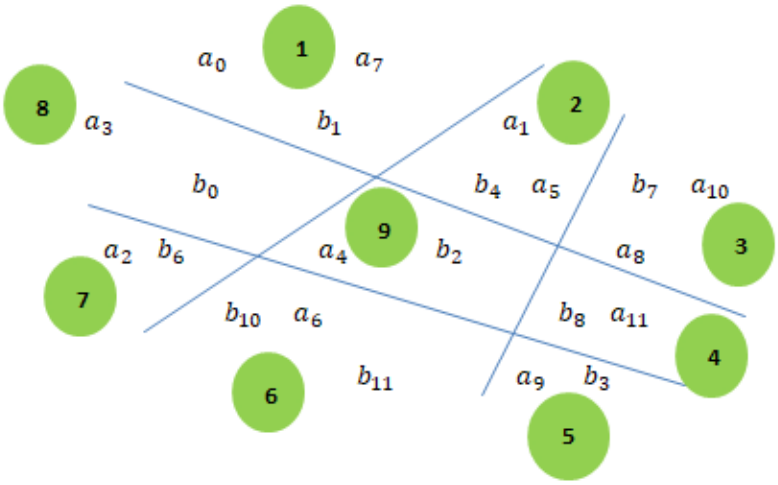

Figure 2: Hard Partitioning

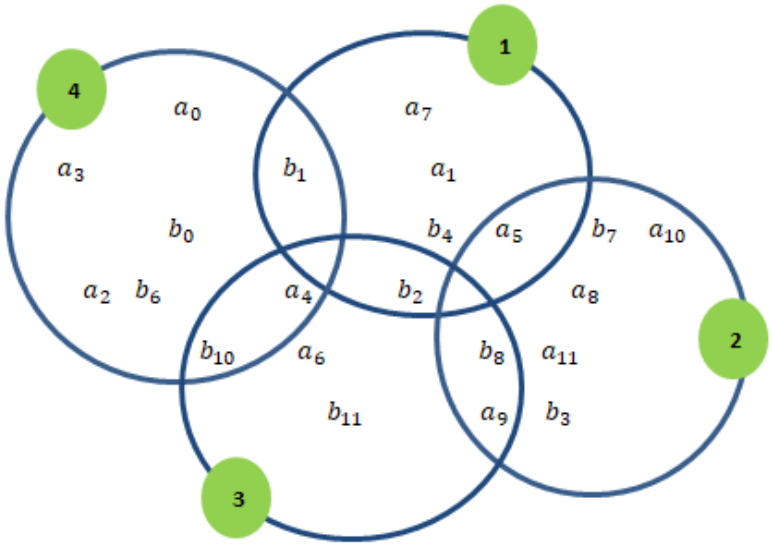

Figure 3: Fuzzy Partitioning

Though the performance of a MPI system depends on the data set on which it operates (field types, number of records, percentage of possible missing value, etc.), configuration of fields and rules, etc.; this system utilizes three measurements to evaluate the indexing stage. These are conducted on test data sets at a staging phase.

(a) It is important to measure the computational complexity of each blocking approach. To achieve this, the metric called as Reduction Ratio (RR) is introduced and it is defined as follows.

$$RR = \frac{e}{l}$$

Where,
e = number of record pairs output by a blocking method

l = |A|×|B|; where A and B are the two data sets used for record linkage process.

(b) To measure how well the blocking algorithm retains the record pairs that are True Matches for further processing, the metric called Pairs completeness (PC) is used and PC is defined as follows.

$$PC = \frac{e_{tp}}{L_{tp}}$$

Where,
etp = Number of true matches in the set of record pairs generated for comparison by the blocking method.

Ltp = Total number of true matches in the dataset

(c) The F-Score is defined as harmonic mean of two variables. Here, F-Score is used to measure comparison between Reduction Ratio and Pairs Completeness. F-score of PC and RR is defined as follows.

$$F - score(PC, RR) = \frac{(2 * PC * RR)}{(PC + RR)}$$

3. Record pair generation

This is covered by the Matching phase of the MPI system. Each input record is sent through the same set of rules mentioned in the previous section in order to find the correct data slice to which the record is closely related to and then the record is matched with the cluster centroids (representative records of each cluster) belonging to the selected data slice. This provides the system the ability to find the small cluster of records which is most closely related to input record. Then using input record with each of the records of the chosen cluster, the system generates a set of records pairs which are the candidate pairs for the next stage. At the end of remaining stages, one of these candidate pairs emerges as the closest match.

4. Field Comparison

This is the fourth stage of record linkage process and is handled by the Matching phase of the this system. From the set of candidate pairs, one pair is considered at a time. A pair is taken and a similarity function (which is set at configuration phase, which may or may not be same for all the fields) is applied to measure how similar the values of a particular field are of the two records in the selected pair. This is done for all the configured fields of the pair. The types of similarity measurement functions available at the configuration phase are described below and these usually belong to one of two major similarity measure types based on the value range of the output of each function; Binary field comparison and Approximate field comparison.

(a) Similarity score is binary (i.e., a value for a record pair of each selected field is drawn from the set [0,1])

(b) Similarity score is continuous (i.e., a real value of a record pair of each selected field is drawn from the range [0,. . . . . ,1])

Table 1: **Application of Similarity Functions.**

Fields:	First Name	Last Name	Gender
Record 1:	Anne	Alexander	Female
Record 2:	Ann	Alexander	Female
Binary Field Comparison:	0	1	1
Approximate Field Comparison:	0.8	1	1

Table 1 illustrates how Binary and Approximation methods are applied on a hypothetical scenario.

The following section briefly describes some of the comparators used in this system. At the configuration phase, these can be linked with the fields of the dataset. (while some of these are recommended for the use on non numeric fields, based on the nature of the data and context, many of these have the potential to be used on any data type).

- n-gram Methods

  Strings are tokenized into n-grams, where n is an integer that can range from 1 to the length of the string. Also, n-grams are generally padded on both ends with (n -1) blank spaces to accurately account for the first and last letters.

- Edit distance

  This method is used with the Levenshtein edit distance. The Levenshtein edit distance between two strings is measured by counting minimal number of character additions, deletions, and substitutions needed in order to transform one string into the other. A dynamic programming approach is used to calculate edit distance so that the minimal number of edits between the strings is iteratively calculated.

- Jaro-Winkler

  Jaro-Winkler distance is used as a string comparator. A modification introduced by Winkler to the Jaro distance mainly identifies errors which are more likely to occur at the end of a string rather than the beginning. Hence, in determining string similarity, greater emphasize is provided for characters at the beginning of the string. Also, the Jaro and Jaro-Winkler metrics is most likely to use for short strings (First name and last name).

- Sequence

  Sequence is used for comparing pairs of sequences of any type. This uses an approach named 'gestalt pattern matching' where the emphasis is mainly not on finding the minimum edit distance but on finding the words which 'looks' similar to humans.

- Metaphone

  In most cases the pronunciation of a certain name, place, etc. could cause ambiguity when entered into a system as text. For example 'schmidt' could be mistaken as 'smith' by the listener. In such cases phonetic algorithms are the best option as the similarity measure and metaphone is one such phonetic algorithm used in our system.
  Metaphone was founded by Lawrence Philips in 1990 and it improves on the Soundex algorithm. Mainly, this is used for indexing words by their English pronunciation.

- Double Metaphone

  Double Metaphone is a modification of the metaphone algorithm by the same author and it was published in June 2000. The modified algorithm is applicable for spelling

variations of any languages whereas the original metaphone algorithm is applicable for only English Language. It is named as double since it results both primary and secondary code for a string. Mainly, this considers both ambiguous words and various surnames written in common ancestry.

- NYSIIS

  New York State Identification and Intelligence System is abbreviated as NYSIIS. The NYSIIS is a phonetic algorithm and it was implemented in 1970.

- New Algorithm of Deep Data Insights

  The 'Deep Data Insight matcher' approach is a new way of name matching where it utilizes encoding, removal of duplicates in coded names and use of a sliding window to match parts of names with each other. This approach enhances the matching score of standard tri-gram if the two names are closer, and retains or decreases the standard tri-gram matching score if the two names are dissimilar. Thus this provides more discrimination between the matches and non matches and also reduces the number of matches that will fall into ambiguous range of matching scores.

As the measurement of performance of each of these, the system utilizes the true positive rate measure.

$$TP_{rate} = \frac{TP}{TP + FP}$$

where

TP = Number of true positives (i.e.Predicted matches that are true positive)

FP = Number of false positives (i.e.Predicted matches that are true non matches)

5. Record Pair comparison

   This falls under the Matching phase of the MPI system. At the configuration phase, weights for each field are set. These weights are selected by either using domain knowledge or running the system on a test data set and calculating the weights based on the results of different trials. Higher weights are given to more significant fields, for example, First name, last name and date of birth fields are more significant than the address or mobile number fields which can change more frequently. The defined weights for each field and similarity scores calculated for each field of each record pair are used to calculate the confidence score for each record pair.

6. Record Pair Classification

   In this step, each record pair is classified as matches, non-matches and ambiguous matches based on confidence score calculated during the above stage and the lower and upper threshold values for confidence score which are set at configuration phase. A new record is considered to be a match if the confidence score is greater than or equal to the upper threshold value and if so a reference is created and if the confidence score is less than or equal to the lower threshold then a new MPI entry is created. These occur during the Matching phase of the the MPI system while the ambiguous records are flagged to be manually inspected during the Ambiguity Resolution phase.

   For measurement of the performance at the end of Record Pair Classification, the system utilizes following measurements on the test data sets.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

where

TP = Number of true positives (Number of predicted matches which are actual matches)

FP = Number of false positives (Number of predicted matches which are actual non-matches)

FN = Number of false negatives (Number of predicted non-matches which are actual non-matches)

3 MPI record retrieval via Search

This MPI solution is supporting standard API endpoints for different tasks. Two of these endpoints are dedicated for searching the MPI. This aspect is utilized by the end users of the system, for example, in the healthcare scenario a desk clerk, nurse or a medical personnel will be searching the system to obtain the patient's data. The search is available in two options.

Text based search : The end user can enter personal information for the mandatory fields and search. The advantage of this text search is that the user can input information with mistakes and still the system will return the closest match. For example in the case of the healthcare scenario if the correct matching patient record in the MPI is containing 'stephen carlson, male, 1970-10-10, ... ' and the desk clerk enters 'steven calsen', the system will still return the correct match (with a less than 100 confidence score).

Facial Recognition : Text based search is error prone, as seen in above example. This system is integrated with a facial recognition based search thus making it more efficient. Here the person's image will be captured in real time by a camera and it will be matched with a repository of images related to MPI records and once the match is found the MPI record will be returned. In order to increase the accuracy, the system is incorporated with a machine learning algorithm to cluster the repository images during the training phase thus reducing the number of comparisons which need to take place in order to recognize the person.

4 Conclusion

Deep Data Insight MPI system is a stand alone system which provides standard API endpoints and high degree of customization which makes it highly flexible and incorporable with other systems with easy. Use of machine learning, image processing together with standard and new algorithms provide the system with higher accuracy and efficiency. This system is capable of being integrated with both small and large scale projects of various industries. It improves interoperability within or throughout organizations. It provides a single view of a person through an integrated, consistent system thus facilitating the users a path to working with a unified index without having to change existing data structures.

5 References

1. Durham, E.A., 2012. A framework for accurate, efficient private record linkage (Doctoral dissertation, Vanderbilt University).

2. De Bruin, J., 2015. Probabilistic record linkage with the Fellegi and Sunter framework: Using probabilistic record linkage to link privacy preserved police and hospital road accident records.

3. Cohen, W., Ravikumar, P. and Fienberg, S., 2003, August. A comparison of string metrics for matching names and records. In Kdd workshop on data cleaning and object consolidation (Vol. 3, pp. 73-78).