# **CREATING MASTER-DATA FROM DATASET USING SIMILARITY-SCORES OF TEXT-FIELDS**

## **INTRODUCTION / MOTIVATION FOR THE USE-CASE**

## The same hospital/site “*JJ Hospital at Mumbai”*, might be reported differently for clinical-trials being conducted across the globe.

## We want to identify a single entity to which other duplicate records can be matched.

## Although there are industry-standard tools available (Informatica, Oracle, SAP, etc.) that can be used with third party collaborators like Dun & Bradstreet to identify the best possible golden-record (asset), this use-case was intended to prove that open-source code and libraries could be leveraged to produce near standard results.

## Within a dataset of n records, we’d have to compare the 1st record with the remaining (n-1) records, the 2nd record with the remaining (n-2) records, and so on. Thus, there would be n\*(n-1)/2 unique combinations to be considered.

## Between 2 different datasets of m and n records each, there would similarly be m\*n unique combinations to be considered.

## **IMPLEMENTATION CHOICES**

## Levenshtein in precompiled C-function, rather than JaroWinkler and Jaccard similarity. \*\*cite the other research paper stating advantages of Levenshtein; \*\*cite the Medium blog that summarizes these algorithms.

## Python for easy data-wrangling.

## R for only computations since in-memory statistical computations are much faster.

## Two main functions inspired from RecordLinkage library (\*\*cite the package) to generate:

## [n\*(n-1)/2] candidates for deduplication within a single dataset, or

## [m\*n] candidates for identifying duplicates between two different datasets

## **ARCHITECTURE**

## Sort the dataset by relevant features so that minibatches contain most of the duplicates already.

## Describe the recursive approach diagram end-to-end from sourcing-data to creating list of csv-files.

## **FUTURE SCOPE**

## Lemmatize each word before match-scores using NLTK (Natural Language Processing ToolKit) in Python.

## Implement an incremental approach to match: incoming dataset of delta-records vs the already identified master-records.

## Implement a better way to identify master-records, than just choosing the very first occurring record amongst the subset of potential duplicates. Possible to choose the golden master-record based on number of common occurrences within that subset of potential duplicates.

## **REFERENCES**

## \*\* the other research paper stating advantages of Levenshtein

## \*\* the Medium blog that summarizes these algorithms

## \*\*cite the RecordLinkage package