Overview of Functions

Data Cleaning Functions

Normalizer Function	Description
AccentRemover	Removes diacritics.
DigitRemover	Removes all digits.
GenderNormalizer	Transforms all gender characters sequences (male, female, m, f, w, 0, 1, …) into an uniform representation.
LetterLowerCaseToNumberNormalizer	Converts o→0, l→1, z→2, q→4, s→5, g→9 to handle OCR errors.
LetterUpperCaseToNumberNormalizer	Converts O→0, L→1, Z→2, A→4, S→5, G→6, B→8 to handle OCR errors.
LowerCaseNormalizer	Converts all letters to lower case.
NonDigitRemover	Remove all characters that are no digits.
NullRemover	Removes “NULL” and “null” strings.
NumberToLetterLowerCaseNormalizer	Converts 0→o, 1→l, 2→z, 4→q, 5→s, 9→g to handle OCR errors.
NumberToLetterUpperCaseNormalizer	Converts 0→O, 1→L, 2→Z, 4→A, 5→S, 6→G, 8→B to handle OCR errors.
PunctuationRemover	Removes punctuation like dots or commas.
SpecialCharacterRemover	Removes special characters like ?,!,$, …
StandardStringNormalizer	Applies WhitespaceRemover, UmlautNormalizer, AccentRemover, PunctuationRemover, NumberToLetterLowerCaseNormalizer, LowerCaseNormalizer.
StandardNumberNormalizer	Applies LetterLowerCaseToNumberNormalizer, LetterUpperCaseToNumberNormalizer, NonDigitRemover.
SubstringNormalizer(0,x)	Extracts the substring from beginning to the x-th character.
TrimNormalizer	Removes any trailing or leading whitespace.
UmlautNormalizer	Converts ä→ae, Ä→Ae, ö→oe, Ö→Oe, ü→ue, Ü→Ue, ß→ss.
UpperCaseNormalizer	Converts all letters to upper case.
WhitespaceRemover	Removes all whitespace.

Extractor Functions

Extractor Function	Description
UnigramExtractor	Constructs all consecutive substrings of length 1. Useful only for few attributes like gender.
BigramExtractor	Constructs all consecutive substrings of length 2. Useful for all attributes.
TrigramExtractor	Constructs all consecutive substrings of length 3. Useful for all attributes. Trigrams tend to lead to higher precision but lower recall than bigrams.
IdentityExtractor	Extracts the given string. Useful for attributes like gender or for error-free attributes.

Hashing Methods

Hashing Method	Description
Double Hashing	Uses two hash functions to simulate a certain number of hash functions.
Random Hashing	Uses a pseudo-random number generator to simulate a certain number of hash functions.

For more details see:
M. Franke et al. Evaluation of Hardening Techniques for Privacy-Preserving Record Linkage. In EDBT, pages 289-300. 2021. https://doi.org/10.5441/002/edbt.2021.26

Blocking Methods

Blocking Method	Description
Hamming LSH	Blocking based on locality-sensitive hashing based on the Hamming distance.
Jaccard LSH	Blocking based on locality-sensitive hashing based on the Jaccard similarity.

For more details see:
M. Franke et al. Parallel Privacy-Preserving Record Linkage Using LSH-Based Blocking. In IoTBDS, pages 195–203. 2018. https://doi.org/10.5220/0006682701950203

Similarity Functions

Similarity Function	Description
Jaccard Similarity	Given two Bloom filters x,y the Jaccard similarity is defined as |x AND y| / |x OR y| where |·| denotes the cardinality (number of 1-bits).
Dice Similarity	Given two Bloom filters x,y the Jaccard similarity is defined as (2*|x AND y|) / (|x| + |y|) where |·| denotes the cardinality (number of 1-bits). The Dice similarity is not a proper distance metric as it does not satisfy the triangle inequality.
Hamming Similarity	Given two Bloom filters x,y the Hamming similarity is defined as 1-(|x XOR y| / Max(|x|, |y|)) where |·| denotes the cardinality (number of 1-bits).

For more details see:
A.P. Brown et. al. Evaluation of approximate comparison methods on Bloom filters for probabilistic linkage. In International Journal of Population Data Scienc. 2018.
https://doi.org/10.23889/ijpds.v4i1.1095

Post-processing Methods

Post-processing Method	Description
Max1-both (Symetric Best Match)	For every record only the best matching record of the other source is accepted.
Stable Marriage (Stable Matching)	Generates a stable matching, where no two records of the two sources both have a higher similarity to each other than to their matching record.
Hungarian Algorithm (Maximum Weight Matching)	Generates a maximum weight matching, where the sum of the overall similarities between records in the final linkage result is maximized.

For more details see:
M. Franke et al. Post-processing methods for high quality privacy-preserving record linkage. In Data Privacy Management, Cryptocurrencies and Blockchain Technology, pages 263–278. Springer, 2018.
https://doi.org/10.1007/978-3-030-00305-0_19