POZNAN UNIVERSITY OF TECHNOLOGY

CAICS
Interdisciplinary Centre for Artificial Intelligence and Cybersecurity

Data deduplication

Robert Wrembel
Poznan University of Technology
Faculty of Computing and Telecommunications, Institute of Computing Science
Interdisciplinary Centre for Artificial Intelligence and Cybersecurity
Poznań, Poland

Outline

CAICS

- **Motivation**
- **Fundamentals of data deduplication**
  - **base-line data deduplication pipeline**
  - **example algorithms**
- **The content is based on research literature**
  - **some surveys include**

- V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, K. Stefanidis: An Overview of End-to-End Entity Resolution for Big Data. ACM Comput. Surv. 53(6), 2021
- G. Papadakis, D. Skoutas, E. Thanos, T. Palpanas: Blocking and Filtering Techniques for Entity Resolution: A Survey. ACM Computing Surveys, 52:(3), 2021
- G. Papadakis, E. Ioannou, T. Palpanas: Entity Resolution: Past, Present and Yet-to-Come. EDBT 2020
- A. Colyer: The morning paper on An overview of end-to-end entity resolution for big data. 2020, https://blog.acolyer.org/2020/12/14/entity-resolution/
- F. Naumann: Similarity measures. Hasso Plattner Institut, 2013
- P. Christen: Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Data-Centric Systems and Applications, Springer 2012, ISBN 978-3-642-31163-5

IT landscape in a FI

- Dozens to over a hundred of data repositories to integrate
  - on premise databases from all software houses (mainly relational)
  - xls files
  - csv files
  - streaming sources
  - ...
- Thousands to dozens of thousands of integration (ETL) processes

Motivating facts

- Financial institutions (FIs)
  - strict data governance policies defined by national and international regulatory bodies
  - use thoroughly tested software
- FIs data are dirty
  - missing values
  - erroneous values
  - outdated values
  - duplicates (mainly customers data)
  - assumed that 1%-5% of records are duplicated

Motivating facts

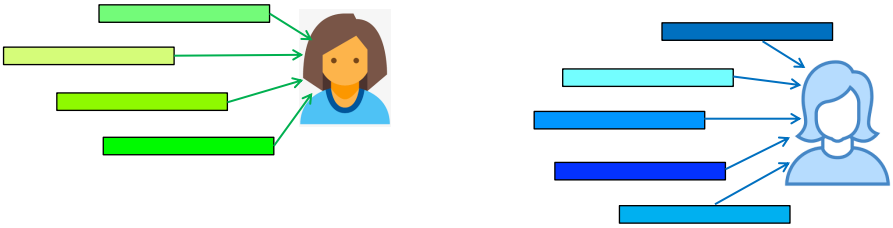
- Sources of duplicate data
  - FI acquisition
  - different bank products require separate customer records
  - imperfection of software used
  - data errors
- Duplicated data cause
  - loss of money
  - deterioration of FI reputation

Deduplication: introduction

- deduplication = entity matching = entity resolution = entity reconciliation = duplicate identification = record linkage = reference reconciliation = data matching = string matching
- Entity Resolution (ER) aims to identify different descriptions (entities, records, objects, data instances) that refer to the same real-world entity
  - appearing either within the same or different data sources
  - when unique entity identifiers are not available
  - https://blog.acolyer.org/2020/12/14/entity-resolution/
- No single best algorithm
  - dedicated algorithms for different domains
  - dedicated algorithms for different data types

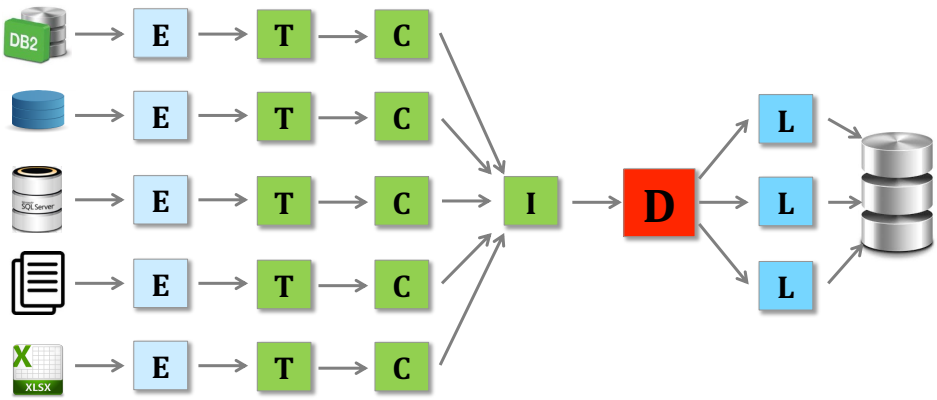
Deduplication: introduction

- Discovering ALL groups of records that describe the same real world object (e.g., customer)



- For large data repositories it is impossible to discover ALL duplicates
- A realistic case: to discover as many duplicates as possible

Deduplication in ETL



Deduplication: introduction

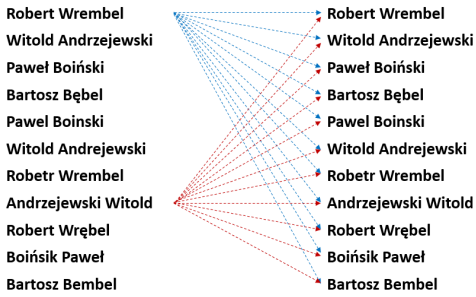
- Before comparing, records should be cleaned
  - homogenizing values (abbreviations, units of measurement, symbols, ...)
  - no special signs, no punctuations
  - no abbreviations
- Problem: how to decide if 2 records represent the same entity?
  - [Wrembel, Robert, ul. Matejki, Poznań]
  - [Wrębel, Robert, ul. Matejki, Poznań]
- Case 1: natural identifiers (e.g., ID, SSN, PESEL, email, mobile#) available
  - but email, mobile# may change in time for the same person
- Case 2: no natural identifiers available
  - approximate/probabilistic decision based on a similarity measure

Deduplication: introduction

- How to compare records?
  - the worst case: each entity compared to all the other entities
    - $O(n^2)$ → deduplication
    - O(n*m) → record linkeage
  - problem: efficiency

Deduplication complexity

- **Deduplication complexity: $O(n^2)$**



- **Bank example: over 20mln customer records**
  - **$2*10^7 * 2*10^7 = 4*10^{14}$ comparisons**
  - **one comparison = 10^{-9}s → comparing the whole set: $4*10^5$s = 4.6 days → must be optimized**

Deduplication: similarities

- **Computing similarities between records in pairs**
  - **computing similarities between attribute values → similarity measures**

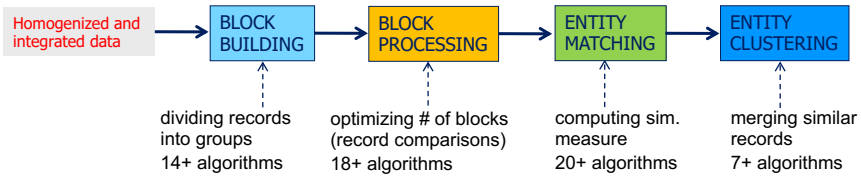
    W. Andrzejewski, B. Bebel, P. Boinski, M. Sienkiewicz, R. Wrembel: Text Similarity Measures in a Data Deduplication Pipeline for Customers Records. DOLAP 2023

  - **computing overall records similarity → similarity classess: T, P, N**
  - **weights of attributes**
  - **similarity thresholds**

	TaxpayerID	Fname	Lname	Country	Title	Domain
	132018	Robert	Wrembel	Poland	Prof.	data wrehouses
sim: 0.7	w1	w2	w3	w4	w5	w6
	TaxpayerID	Fname	Lname	Country	Title	Domain
	132018	Robert	Wrębel	PL	professor	data integration

Deduplication pipeline

- **Improving performance**
  - **avoiding N^2 comparisons of records**
  - **finding groups of similar records**
- **Base-line deduplication pipeline (BLDDP)**



- G. Papadakis, L. Tsekouras, E. Thanos, G. Giannakopoulos, T. Palpanas, M. Koubarakis: Domain- and Structure-Agnostic End-to-End Entity Resolution with JedAI. SIGMOD Record, Vol. 48, No. 4, 2019
- G. Papadakis, D. Skoutas, E. Thanos, T. Palpanas: Blocking and Filtering Techniques for Entity Resolution: A Survey. ACM Computing Surveys, 52:(3), 2021

BLDDP

- **Complex algorithms (concepts and mathematical models)**
- **Input data must be cleaned beforehand**
- **Verified on either data sets generated artificially or small real data sets, e.g.:**
  - **restaurants - 500 rec**
  - **products - from 1000 to 22000 rec**
  - **movies - 23000 rec**
  - **bibliographical - from 2600 to 60000 rec**
  - **patients - 128000 rec**
  - **synthetic - from 10000 to 300000 rec**

data set characteristics

- L. Gagliardelli, G. Papadakis, G. Simonini, S. Bergamaschi, T. Palpanas: GSM: A generalized approach to Supervised Meta-blocking for scalable entity resolution. Inf. Syst. 120: 102307 (2024)

Block building

- a.k.a. blocking, indexing
- Goal: to eliminate not necessary record comparisons without discarding any matching entities
  - based on values of entity features (a.k.a. blocking keys) it organizes similar entities into the same block, then comparisons are made within a given block

Blocking

- Types
  - disjoint blocking: a record is included in only one block
  - overlapping blocking: a record may be included in multiple blocks → probability of finding a better match is higher at a cost of more record comparisons
- Types
  - schema aware block building
    - values of semantically equivalent attributes are compared
  - schema agnostic block building
    - values of all attributes are compared
- Challenge: selecting the right blocking key(s)

Blocking

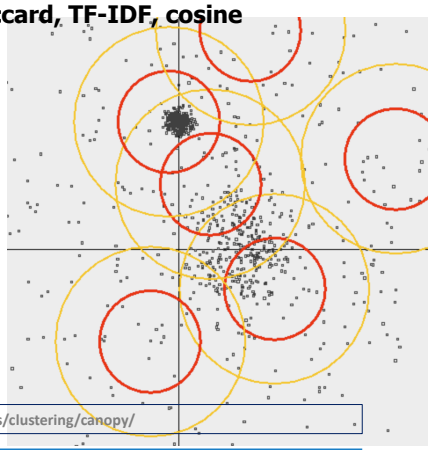
Canopy clustering

- computationally cheap algorithm for pre-clustering for more accurate algorithms (k-means, dbscan)
- two distances are used
  - T_L loose distance
  - T_T tight distance
  - $T_L > T_T$
- 1. randomly select a center point p^i_C and create canopy C_i where p^i_C is the center point of C_i
- 2. assign every point to C_i if the distance between a given point and $p^i_C < T_L$
  - points within circles of radius T_T cannot be center points in the next iteration
- 3. repeat step 1-2 until there are no more data points

Blocking

Canopy clustering

- create n-dimensional overlapping clusters - blocks of records
- distance (similarity): often Jaccard, TF-IDF, cosine



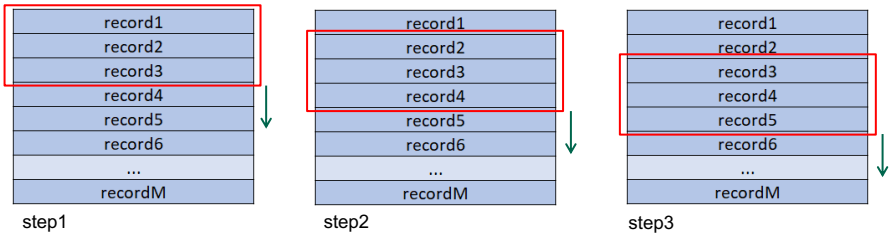
- Figure from: https://mahout.apache.org/docs/latest/algorithms/clustering/canopy/

Blocking

- **Hashing**
  - on some attributes (hash keys) → O(n)
  - drawback: records must have identical values of hash keys to hash into the same bucket → applicable for exact matching

Blocking

- **Sorted neighbourhood**
  - sort records by a given attribute (list of attributes)
  - assumption: more similar records are located closer to each other
  - compare records only within a moving window of N records
  - within the window all with all are compared
  - open issue: the size of the window



Blocking

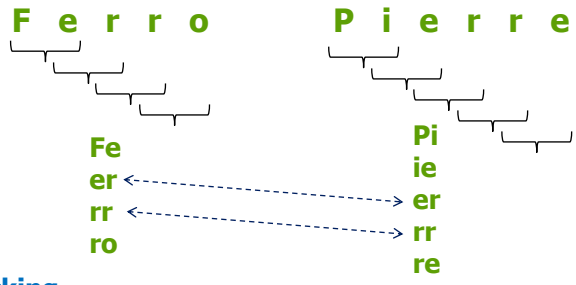
- Token-based → redundancy positive blocks

r1: {name: Robert Wrembel, degree: prof.}
r2: {name: Robert Wrębel, degree: professor}
r3: {name: Witold Andrzejewski, degree: dr}
r4: {name: Witold Andrzejewski, degree: dr inż.}
r5: {name: Paweł Boiński, degree: dr}
r6: {name: Bartosz Bębel, degree: dr}

block fname (Robert): {r1, r2}
block fname (Witold): {r3, r4}
block fname (W): {r4}
...
block lname (Andrzejewski): {r3, r4}
block degre (dr): {r3, r5, r6}

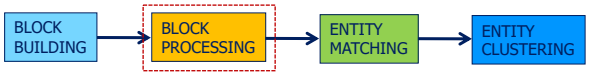
Blocking

- bi-gram example



- n-gram based blocking
  - group similar records together based on n-grams they share
  - records that share a parameterized minimum number of n-grams are grouped into the same block
- n-grams can be organized into an inverted index
  - all records that contain the same token in their blocking key reside in the same inverted index list

Block processing



- **Goal: to minimize the number of comparisons, i.e., to discard unnecessary comparisons:**
  - **between blocks (in case of overlapping blocks)**
  - **between records in a given block**

Block processing

- **Block purging**
  - **applies to oversized blocks that correspond to tokens of little discriminativeness → records in such blocks may not represent duplicates**
    - **e.g., a block build on the first 3 letters of a city name**
  - **executed based on heuristics that identify oversized blocks by estimating an upper limit on the number of matches a block is expected to contain**
  - **the heuristics are designed with the support of experiments**

- G. Papadakis, E. Ioannou, C. Niederée, P. Fankhauser: Efficient entity resolution for large heterogeneous information spaces. WSDM, 2011

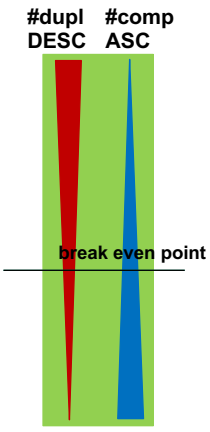
Block processing

- **Duplicate propagation**
  - as blocks are overlapping, the same objects are compared more than once → avoiding this by propagating the identified matches to the subsequently processed blocks
  - a central hash table contains all the matches detected so far
  - before comparing a pair of objects, check whether any of them is registered in the hash table → if true for at least one of them then skip the comparison

- G. Papadakis, E. Ioannou, C. Niederée, P. Fankhauser: Efficient entity resolution for large heterogeneous information spaces. WSDM, 2011

Block processing

- **Block pruning**
  - order blocks by
    - expected number of duplicates → descending
    - expected number of comparisons → ascending
    - blocks with high number of duplicates and low number of comparisons go up
  - the lower the ranking positions the fewer duplicates can be discovered at a higher comparison cost
  - apply a formula to define a break even point where the possibility of finding additional matches is too expensive
  - block pruning → discarding blocks below the break even point from processing



- G. Papadakis, E. Ioannou, C. Niederée, P. Fankhauser: Efficient entity resolution for large heterogeneous information spaces. WSDM, 2011

Block processing

- **Block filtering**
  - **importance of a block for a given object o_i = maximum number of blocks o_i resides in → count block assignments per object**
  - **remove o_i from the least important block(s)**

- G. Papadakis, G. Papastefanatos, T. Palpanas, M. Koubarakis: Scaling Entity Resolution to Large, Heterogeneous Data with Enhanced Meta-blocking. EDBT, 2016

Block processing

- **Split large blocks into smaller (below max allowed size) [1]**
- **Merge blocks with similar blocking keys (similarity threshold) [2]**

1. A. Das Sarma, A. Jain, A. Machanavajjhala, P. Bohannon: An automatic blocking mechanism for large-scale de-duplication tasks. CIKM, 2012
2. J. Fisher, P. Christen, Q. Wang, E. Rahm: A Clustering-Based Framework to Control Block Sizes for Entity Resolution. KDD, 2015

Block processing

- Size-based block clustering
  - strategies
    - merging small blocks that correspond to similar blocking keys
    - splitting large blocks into smaller ones
  - to balance block sizes → balancing parallel processing of record matchings in blocks

- J. Fisher, P. Christen, Q. Wang, E. Rahm: A Clustering-Based Framework to Control Block Sizes for Entity Resolution. KDD, 2015

Block processing

- Iterative blocking
  - whenever a new pair of duplicates is detected (r_i,r_m) their descriptions are merged → r_{im}
  - r_{im} replaces (r_i,r_m) in all blocks
  - compare r_{im} to other records in its blocks and merge if matching
  - repeat until no matches can be found

- S.E. Whang, D. Menestrina, G. Koutrika, M. Theobald, H. Garcia-Molina: Entity resolution with iterative blocking. SIGMOD, 2009

Block processing

- **Meta-blocking**
  - uses a graph to represent comparisons
  - eliminates the same comparisons in multiple blocks
  - uses labels of graph edges to eliminate comparisons below certain threshold
    - methods for computing the values of the labels

- G. Papadakis, G. Koutrika, T. Palpanas, W. Nejdl: Meta-Blocking: Taking Entity Resolution to the Next Level. IEEE Trans. Knowl. Data Eng. 26(8), 2014

Meta blocking: example

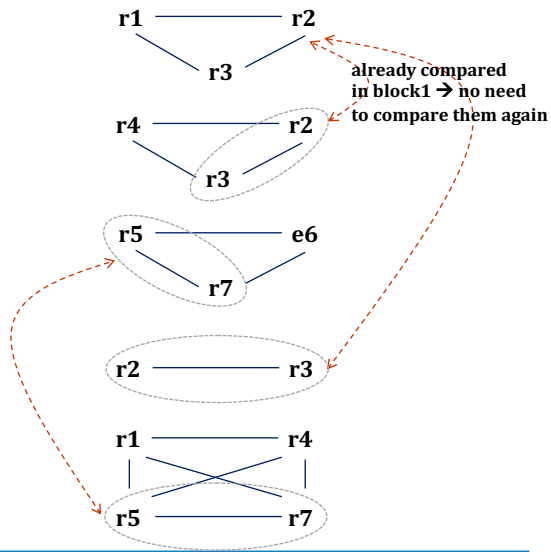
block1: {r1, r2, r3}

block2: {r2, r3, r4}

block3: {r5, r6, r7}

block4: {r2, r3 }

block5: {r1, r4, r5, r7}



Meta blocking: example

block1: {r1, r2, r3}

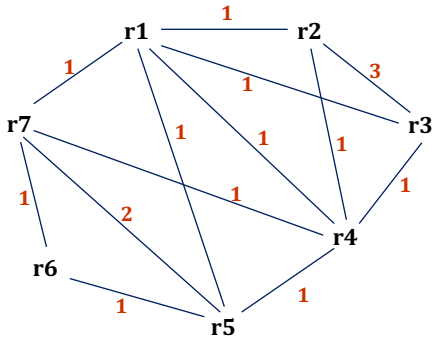
block2: {r2, r3, r4}

block3: {r5, r6, r7}

block4: {r2, r3 }

block5: {r1, r4, r5, r7}

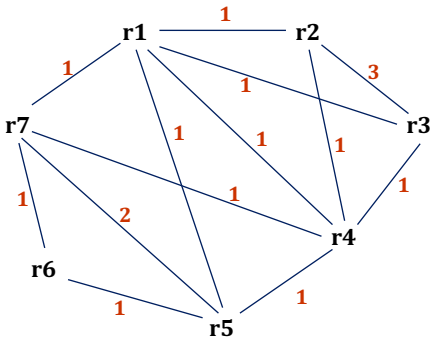
edge weight: #blocks where e_i and e_m are compared

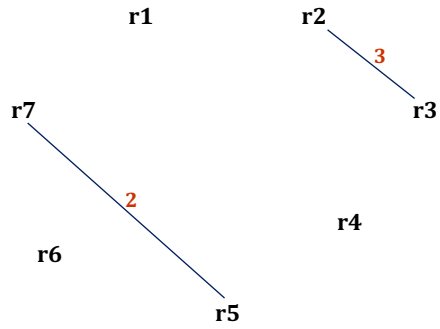


trade off: lower number of entity comparisons at a cost of lower recall
recall: TP/(TP+FN)
FN: entities that represent the same object but were not discovered as being the same

Edge prunning

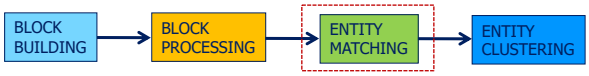
- Remove edges that are below avg weight [1]
- Retain top k weighted wedges [1, 2]





1. G. Papadakis, G. Koutrika, T. Palpanas, W. Nejdl: Meta-Blocking: Taking Entity Resolution to the Next Level. IEEE Trans. Knowl. Data Eng. 26(8), 2014
2. F. Zhang, Z. Gao, K. Niu: pruning algorithm for Meta-blocking based on cumulative weight. Journal of Physics: Conference Series, 887, 2017

Entity matching



- Goal: to assign a label to each pair of entities being compared
  - {match, non-match}
  - {match, probably match, non-match}
- A similarity measure (function) is used for computing similarities between compared entities

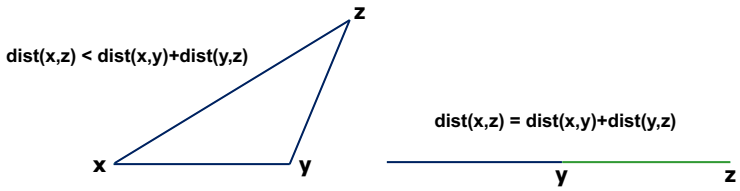
- M. Alamuri, B.R. Surampudi, A. Negi: A survey of distance/similarity measures for categorical data. Int. Joint Conf. on Neural Networks (IJCNN), IEEE, 2014
- S. Boriah, V. Chandola, V. Kumar: Similarity Measures for Categorical Data: A Comparative Evaluation. SIAM Int. Conf. on Data Mining (SDM), 2008
- P. Christen. 2012. Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, 2012
- P. Christen: A Comparison of Personal Name Matching: Techniques and Practical Issues. Int. Conf. on Data Mining (ICDM), IEEE, 2006
- F. Naumann: Similarity measures. Hasso Plattner Institut, 2013
- M. del Pilar Angeles, A. Espino-Gamez. 2015. Comparison of Methods Hamming Distance, Jaro, and Monge-Elkan. Int. Conf. on Advances in Databases, Knowledge, and Data Applications, 2105

Entity matching

- Matching uses similarity function $sim(r_i,r_j)$ that maps each pair of records (r_i,r_j) to a similarity value
  - φ measures how similar r_i and r_j are
- Matching (variant 1)
  - matching: $sim(r_i,r_j) >= v$
  - not matching $sim(r_i,r_j) < v$
- Matching (variant 2)
  - not matching: $sim(r_i,r_j) < v_1$
  - unknown: $v_1 <= sim(r_i,r_j) < v_2$
  - matching: $sim(r_i,r_j) >= v_2$

Distance and similarity measure

- A distance is metric when it has the following properties
  - reflexive: distance between string x and x = 0
  - symmetric: distance between string x and y = distance between y and x
  - distance (x, y)>=0
  - triangular inequality
    - dist(x,z)<=dist(x,y)+dist(y,z)



- From distance to similarity measure
  - typically similarity measure = 1 - distance

Similarity measures for text data

- Over 30 different similarity measures for text data
- Phonetic encoding of a text value
  - e.g., Soundex, Phonex, Phonix, NYSIIS, Metaphone
- Edit distance - count the smallest number of edit operations that are required to convert string s1 into s2
  - e.g., Levenshtein, Damerau-Levenshtein, Smith-Waterman
- N-grams - split compared strings into n-character sub-strings (n-grams) and count common n-grams
- Set similarity - a similarity depends on the number of common elements in compared sets
  - e.g., Overlap, Jaccard, Sorensen-dice

Similarity measures for text data

- The longest common sub-sequence or the longest common sub-string
- Vector representation in an m-dimensional space
  - e.g., TFIDF, Cosine
- Compression techniques
  - e.g., BZ2, Lempel-Ziv-Mrkov, ZLib
- Ensemble of measures
  - e.g., Monge-Elkan + Damerau-Levenshtein + n-gram, Cosine+n-gram

Example

- Different similarity measures produce different values for the same compared strings

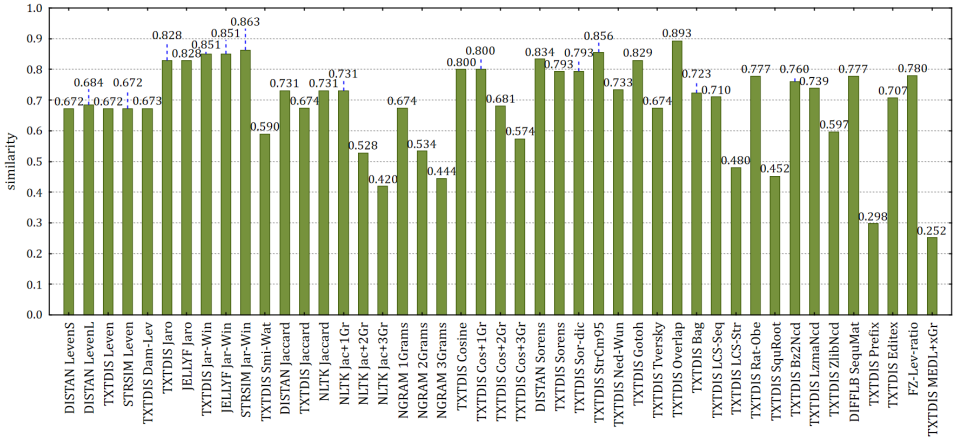
r1	Robert	Wrembel	Wyspiańskiego
r2	Robret	Wręble	Wyspiankiego
Jaro-Winkler	0.961	0.826	0.953
Levenshtein	0.667	0.429	0.846
Overlap	1.000	0.833	0.917

AVG(simJW(r1,r2))=(0.961+0.826+0.953)/3=0.913 ← similar
AVG(simLe(r1,r2))=(0.667+0.429+0.846)/3=0.647 ← probably similar
AVG(simOv(r1,r2))=(1+0.833+0.917)/3=0.916 ← similar

Example

	are they similar?	
B. CHROBREGO	<---------->	BOLERSŁAWA CHROBREGO
B CHROBREGO	<---------->	BOLESLAWA CHROBREGO
B CHROBREGO 10	<---------->	BOLESŁ. CHROBREGO
B CHROBREGO 42B		BOLESŁAW CHROBREGO
B,CHROBREGO		BOLESŁAWA CHROBREGO
B.CHROBREGO		BOLESŁAWA CHROBREGO
B.CHROBREGO 33/72		BOLESŁ.CHROBREGO
B.CHROBREGO OŁOBOK		BOLESŁWA CHROBREGO
B.CHROBREGO SKR.60		BOOESŁAWA CHROBREGO
BOL. CHROBREGO		CHROBREGO
BOL CHROBREGO		CHROBREGO 10
.BOL.CHROBREGO		CHROBREGO 22A/6
BOL.CHROBREGO		CHROBREGO KOWALEW
BOL.CHROBREGO OŁOBOK		CHROBREGO PAŃSTWOWY DOM D

Similarity measures



- **Which similarity measure is the best for a given deduplication problem?**

Semantic relationships augmenting similarities

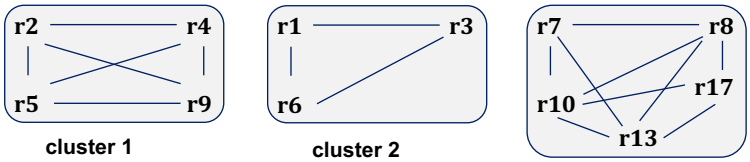
- Context-based similarity (semantic relationships)
  - data source S1 (healthcare): object patient100
  - data source S2 (healthcare): object person131
  - data source S3 (banking): object customer456
  - patient100 more similar to person131 since both exist in the same healthcare context

Entity clustering

- Compared pairs of records with their similarities are merged into graphs
  - nodes: records
  - edges: similarity values
- Goal: find sub-graphs (clusters) of records that represent the same real-world object (with a given high probability)
  - prune edges that connect non-duplicates
  - infer indirect edges between records

Entity clustering

- **Multiple clustering algorithms**
  - **use the similarity measure between records**



cluster 1
φ > sim_value

cluster 2
φ > sim_value

cluster 3
φ > sim_value

Entity merging

- **Merge semantically identical records in a cluster into one final augmented record**

TaxpayerID	Fname	Lname	Country	Title	Domain
132018	Robert	Wrembel	Poland	Prof.	data wrehouses

+

TaxpayerID	1sName	2ndName	Last_Name	Education	City	Born
132018	Robert	Andrew	Wrembel	univ.	Poznań	1968

TaxpayerID	1sName	2ndName	Last_Name	Education	City	Born	TaxpayerID	Country	Title	Domain
132018	Robert	Andrew	Wrembel	univ.	Poznań	1968	132018	Poland	Prof.	data wrehouses

Deduplication pipeline: challenges

- **C1: which algorithms to apply to a given data set? → selecting the best combination of the algorithms**
  - **deduplication quality (precision, recall)**
  - **execution costs**
  - **35280+ combinations of these algorithms**
  - **searching the full space of alg. combinations is impossible**
  - **techniques for pruning search space**
    - **some methods may be better for deduplicating personal data, some for bibliographical data, some for addresses**
    - **knowledge of a domain expert is crucial**
  - **an automatic approach does not exist**

Deduplication pipeline: challenges

- **C2: how to figure out adequate similarity measures to be used for a given deduplication problem at hand?**
  - **experiments**
- **C3: how to figure out adequate weights for attributes being compared?**
  - **expert knowledge**
  - **learning methods**
- **C4: how to figure out adequate similarity thresholds between classes: duplicates, probably duplicates, not duplicates?**
  - **expert knowledge**
  - **learning methods**

ML for deduplication

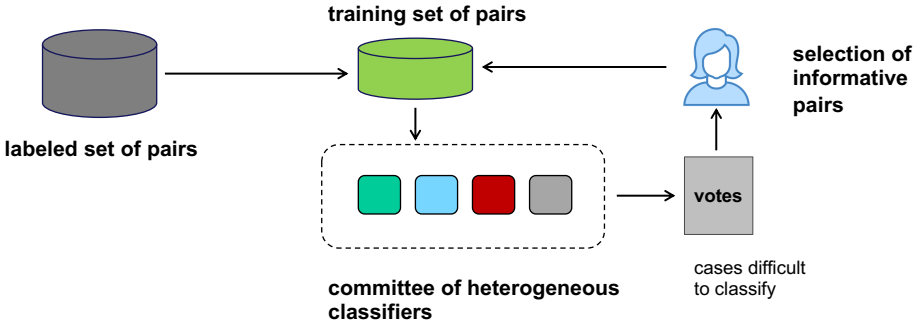
- Problem
  - large set of training data is needed
  - labeling record pairs is a time consuming task
- Standard solution → classification
  - classes: duplicates (T), non-duplicates (F)
  - classes: duplicates (T), probable-duplicates (P), non-duplicates (F)

ML for deduplication: Active Learning

- Alternative solution → Active Learning
  - iterative process
  - at each iteration uncertain pairs are labeled manually by experts
  - at each iteration only the most informative pairs (the most difficult to classify) are labeled
  - expert labeled pairs are added into the training data for the next iteration of retraining a classifier
  - a stop criterion for the number of iterations is needed

ML for deduplication: Active Learning

- Active Learning



ML for deduplication: Active Learning

- How to choose the most informative pairs
  - query by committee: pairs that receive the highest number of disagreement from classifiers
- The committee of classifiers should be heterogeneous
- Classes are imbalanced → much more pairs in class F than in T
  - right sampling strategy, e.g., stratified sampling

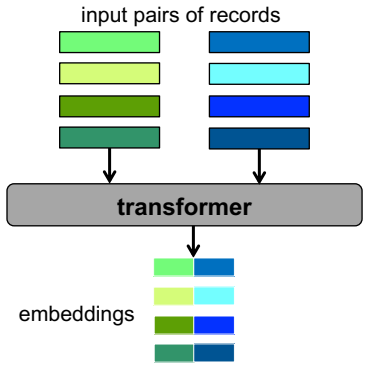
ML for deduplication (1)

- HeALER
- Active learning approach
- Committee of heterogeneous classifiers
  - SVM, one-vs-rest logistic regression, logistic regression, decision tree
- Data sets
  - ACM-DBLP (4800+ records)
  - GScholar-DBLP (66000+ records)

- X. Chen, Y. Xu, D. Broneske, G. Campero Durand, R. Zoun, G. Saake: Heterogeneous Committee-Based Active Learning for Entity Resolution (HeALER). ADBIS, 2019

Deep Learning for deduplication

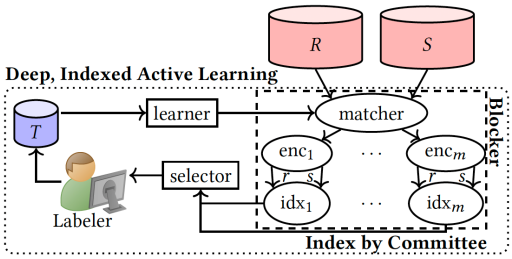
- Transformer
  - trained to encode input pairs of records into multi-dimensional numerical vectors ⟶ embeddings
  - a similarity function is applied to the vectors and constitutes a training dataset for a binary classifier
  - implemented as a recurrent neural network (RNN)
  - once trained on a large corpus, the transformer is able to learn the semantics of words
  - embeddings are typically indexed for fast similarity search
- Claim: RNN can be trained on large external knowledge bases to reduce the need for annotated data



ML for deduplication (2)

- **Characteristics**
  - **active learning approach**
  - **integrated matcher and blocker**
  - **deep neural network used to build matcher and blocker**
- **Blocker creates blocks of possible duplicates → detects duplicates based on embedding similarity**

the picture was taken from the paper



- A. Jain, S. Sarawagi, P. Sen: Deep Indexed Active Learning for Matching Heterogeneous Entity Representations. Proc. VLDB Endow. 15(1), 2021

ML for deduplication (2)

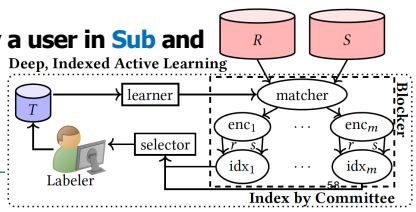
- **Matcher uses transformer-based pretrained language models (TPLM, e.g., BERT, RoBERTa)**
- **For each pair of records (r, s) Matcher assigns a probability of the pair representing a duplicate**
- **Matcher uses Transformer to get a joint embedding E(r, s) of pair (r, s)**

ML for deduplication (2)

- Transformer receives concatenated tokens of record pairs
  - [start], $r_1 \ldots r_n$, [separator], $s_1 \ldots s_m$, [separator]
  - multiple encoders $enc_1, \ldots enc_m$ for creating embeddings
  - transformer assigns for each token an embedding that represents its semantics in the context of the current record
  - the contextual embeddings can eliminate problems with spelling mistakes and different abbreviations
  - embeddings are indexed for faster retrieval

ML for deduplication (2)

- Start with an initial set T of labeled (duplicates, non-duplicates) record pairs
- Iteratively collect new labeled pairs by the active learning loop
- In each iteration of the loop execute steps
  - train the Matcher model that given a pair of records it can assign a probability of the pair being a duplicate
  - train the Blocker model to encode records in R or S so that duplicates are co-located close to each other
  - perform an indexed nearest neighbor search over the embeddings to find a candidate set of likely duplicate pairs
  - select a subset Sub of the candidate set using probability labels from Matcher
  - collect labels on pairs assigned by a user in Sub and augment set T



© R.Wrembel (PUT and CAICS, Poland)

ML for deduplication (2)

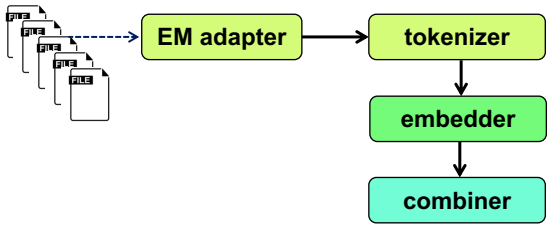
- **Data sets in experiments**
  - **Walmart, Amazon, Google**
  - **DBLP, ACM, GScholar**
  - **max. number of records: ~67000**

AutoML and DL for deduplication

- **AutoML → automatic identification of efficient ML algorithms for minimizing human involvement and taking into consideration some constraints**
  - **limited computational budget (typically CPU and RAM)**
  - **formalized as a Combined Algorithm Selection and Hyperparameter optimization (CASH) problem**
- **Goal**
  - **to select the most suitable classification algorithms (classes T, N)**
  - **to tune hyperparameters**
    - KNN: the number of neighbors
    - SVM: kernel (linear, polynomial)
    - random forest: number of random features to sample at each split point

ML for deduplication (3)

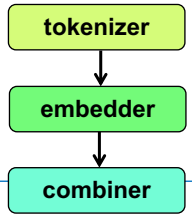
- AutoML + Deep Learning
- EM adapter: encodes input records for further training of the AutoML system
  - tokenizer
  - embedder
  - combiner



- M. Paganelli, F. Del Buono, M. Pevarello, F. Guerra, M. Vincini: Automated Machine Learning for Entity Matching Tasks. EDBT, 2021

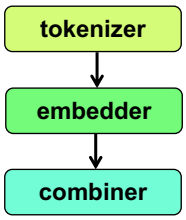
ML for deduplication (3)

- Tokenizer
  - transforms a pair of records (r1, r2) represented as
    a^{r1}_1, ..., a^{r1}_2, a^{r2}_1, ..., a^{r2}_m
    into one or more token sequences
  - modes of creating tokens
    - attributes are concatenated into a unique sentence (a schema is lost)
    - the values of the same attribute in a pair of records are coupled → multiple sub-pairs, one for each attribute, are created
    - some attributes are tokenized by mode 1 and some by mode 2 → hybrid



ML for deduplication (3)

- **Embedder**
  - encodes a token sequence into a multi-dimensional vector → embedding
  - typically, a pre-trained word embedding NN is used
    - in the paper the following embedders were applied: Bert, DistilBert, Albert, Roberta, XLNET
- **Combiner**
  - as multiple embeddings can be generated for the same pair of input records, they must be aggregated into a single multi-dimensional vector



ML for deduplication (3)

- **Data sets in experiments**
  - Walmart, Amazon, Google
  - DBLP, ACM, GScholar
  - max. number of records: ~28000

ML for deduplication: challenges

- C5: how to construct a learning data set of a reasonable size for 20M customers' database?
  - manual tagging of 1000 pairs of rows
    - 3 persons, including a bank expert
    - tagging throughput: 1.33 rec/min
    - 22.2h for tagging 1000 rows
  - conclusion: impossible to construct a learning data set manually

ML for deduplication: challenges

- C6: how to efficiently verify a model on dozens of thousands of records?

- Rules for deduplication
- C7: large data sets may need hundreds of rules → manual management of such a set of rules is impossible
  - rules subsumption detection
  - rules firing order optimization
    - e.g., if one rule says not duplicates, no need to fire other rules
- C8: rules visualization

Requirements = reality

- A project must finish within a given monetary budget and time
- Processed data must be accessible either from a relational database or from files (csv, xls)
- Developed algorithms and models must be intuitive, understandable and easy to implement by a technical IT staff at a company
- Ideally the developed methods should be based on out-of-the-box software components

Requirements = reality

- Input data are partially dirty
- Why partially dirty data?
  - first and last names cannot be cleaned without a customer permission (by law)
  - not all data can be cleaned automatically → manual cleaning
    - e.g., only some addresses can be cleaned automatically
  - it is impossible to clean +20M of customer records within a limited time → too many records needs manual cleaning

Requirements = reality

- The developed methods must be efficient → they will be applied to several millions of rows
- The solution must be deployable in the IT architecture used by a company (specific hardware and software) → either in a database or in a standard data science environment
- Only licensed/certified software can be used (especially in financial institutions)