Identification of Real-World Objects in Multiple Databases

Mattis Neiling

Technische Universität Berlin, mneiling@cs.tu-berlin.de

Abstract. Object identification is an important issue for integration of data from different sources. The identification task is complicated, if no global and consistent identifier is shared by the sources. Then, object identification can only be performed through the *identifying information*, the objects data provides itself. Unfortunately real-world data is dirty, hence identification mechanisms like *natural keys* fail mostly — we have to take care of the variations and errors of the data. Consequently, object identification can no more be guaranteed to be fault-free. Several methods tackle the object identification problem, e.g. *Record Linkage*, or the *Sorted Neighborhood Method.*

Based on a novel object identification framework, we assessed data quality and evaluated different methods on real data. One main result is that scalability is determined by the applied preselection technique and the usage of efficient data structures. As another result we can state that *Decision Tree Induction* achieves better correctness and is more robust than *Record Linkage.*

Keywords

DUPLICATE DETECTION, MERGE-PURGE, DATA CLEANSING

1 Introduction

Assumed that information from several databases shall be merged on the entity level, then the information referring to the same real-world objects have to be identified and put together. But often no unique identifiers are available from the sources such as the *Social Insurance Number SSN* for American residents or the *International Standard Book Number ISBN* for print media. In this situation one has to use the identifying information available from the sources however reliable or correct they may be.

Previous publications the author contributed to stressed the importance of a generic *framework for object identification*, e.g. Neiling and Jurk (2003).As result of our research, we developed a generic object identification framework, mainly consisting of three successive steps: Conversion, Comparison, and Classification. In addition, the framework covers: (1) concepts for identification, (2) its software architecture, (3) data quality characteristics, (4) a preselection technique that ensures efficiency for large databases (incorporating suitable index structures), and (5) a prescription for evaluation, sampling and quality criteria. Based on the framework, an evaluation of different methods of object identification became attainable. We applied extensive benchmarking of several methods on different real-world databases. The framework is described in Neiling and Lenz (2004) in the context of the next German Census that will be basically an *Administrative Record Census*. In this contribution, we will not review all the details of the framework, instead we emphasize on data quality analysis, preselection, and sampling.

The paper is structured as follows: After a review of historical developments we scetch the general model in section 3. After discussing data quality in section 4, we introduce preselection techniques in section 5. Within section 6 we present results of our evaluation. We conclude with a short summary and give an outlook towards further investigations.

2 Historical Development

Starting in the fifties of the last century, a methodology of *Record Linkage* was developed in the sixties, which was continuously improved up to now. It was successfully applied to personal information, mainly for statistical purpose like census data and patient information. The research in this area was mainly focused on the improvement of the underlying *Likelihood-Ratio Test*, without any consideration of alternative methods such as machine learning algorithms. Independent from that development, however, duplicate detection had gotten more and more attention by database researchers in the nineties. Their investigations were performance-driven — computational efficiency was their main goal. Until the end of the last century both approaches to object identification can be said to be complementary —both communities treated it with *different tongues*. Both research directions influenced one another with the beginning of the twenty-first century. Eventually, a methodology could be founded which considers *both* computational and statistical efficiency at the one hand, and the use of learning algorithms on the other one. In our work, we performed an exhaustive comparison of different learning methods.

Record Linkage. Inspired by the work of H. Newcombe et al. (1959), the well-known model for Record Linkage was founded by I.P.Fellegi and A.B. Sunter (1969). Until now, the methodology was continuously enhanced, cf. the proceedings of the two workshops: Kilss and Alvey (1985), and Alvey and Jamerson (1997). For instance, the estimation of the multinomial distribution could be improved by means of variants of the *EM-Algorithm*, cf. Meng and Rubin (1993), Winkler (1993), Liu and Rubin (1994), and Yancey (2002). Further, powerful software packages were developed, cf. Winkler (2001), Bell and Sethi (2001), and Christen et al. (2004). A general overview about the state of Record Linkage can be found in Winkler (1999) and in Gu et al. (2003).

Computational feasibility was a less investigated aspect of Record Linkage, only simple *Blocking* methods were used. Recently, other approaches, e.g.

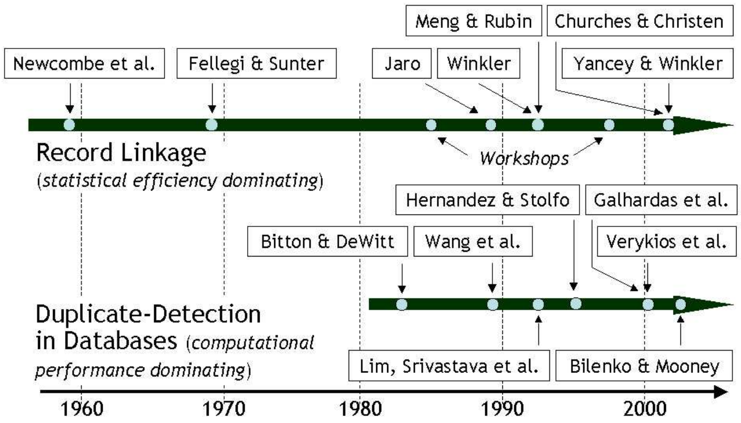

Fig. 1. Overview of Historical Development

clustering are applied, cf. Baxter et al. (2003). Database management systems with their powerful indexes were not investigated — Record Linkage was mostly performed on plain files.

Duplicate Detection in Databases. The start of research on duplicate detection in databases was the seminal work of Bitton and DeWitt (1983) dealing with the removal of identical rows. Wang and Madnick (1989) discussed the identification problem for multiple databases first. Hernandez and Stolfo (1995) invented the *Sorted-Neighborhood-Method*, which is widely used for de-duplication.
Until the end of the twentieth century, there was no use of machine learning algorithms. Recently, many researchers applied supervised learning methods like decision tree induction successfully to object identification, e.g. Neiling and Lenz (2000), Elveky et al. (2002), and Bilenko and Mooney (2003).

3 The General Model for Object Identification

The identification procedure was introduced by Neiling and Lenz (2000) and refined by Neiling and Jurk (2003) and works as follows:

1. *Conversion*: The identifying information are extracted from the original data for each element (e.g. records) and standardized.
2. *Pair Construction and Comparison*: Pairs of elements are built (at least virtually) that fulfill given preselection predicates, cf. section 5. The pairs are compared with sophisticated functions like Minimum-Edit-Distance, N-Gram Distances etc., or simply with comparison patterns for equal/missing/nonequal values.

3. *Classification*: Each comparion vector in the multi-dimensional comparison space is classified by a decicion rule δ w.r.t. a previously induced decision rule as *matched* or *non-matched* (possibly equipped with a score).

The *classifier* δ can be defined manually (e.g. the decision rules of the *Sorted Neighborhood Method*, cf. Hernandez (1996)). Alternatively it can be learned from given example data, i.e. a set of matches and non-matches. For example, within the *Record Linkage* method, the likelihood ratios $\lambda : V \to \mathbb{R}_{\geq 0}$ (the so-called *odds*) are estimated and used as classifier for comparison vectors $v \in V$:

$$\lambda(v) = \frac{P\big(v \mid (a,b) \text{ is matched}\big)}{P\big(v \mid (a,b) \text{ is non-matched}\big)}.$$

Large values $\lambda(v)$ indicate matches, while small values indicate non-matches, whereby values around 1 indicate neither of both. Given predefined error levels of misclassifications, decision bounds $\lambda_l \leq \lambda_u$ can be derived, while pairs with $\lambda(v) \in [\lambda_l, \lambda_u]$ are left unclassified for screening, cf. Fellegi and Sunter (1969). Similarly, if any other classifier provides a score, the error rates could be controlled. This is an important feature, since the costs caused by a misclassification of a match are typically higher than vice versa.

There are many suitable classification methods in literature, e.g. *Decision Tree Induction*, *k-Nearest Neighbor Classification*, *Support Vector Machines*, *Neural Networks*, *Bayes Classifier*, etc. The interested reader may consult textbooks about *Machine Learning* (e.g. Michie et al. (1994), or Berthold and Hand (1999)), or existing *Classification* and *Data Mining Software*. Obviously, the scales of the comparison space has to be considered for a choice. For instance, Record Linkage has been designed for a finite set of nominal values, thus ordinal scaled values are treated as nominal with loss of information. On the other hand, decision tree learner can even deal with mixed scales and are therefore well-suited. Multiple combinations of classifiers has been studied by Tejada et al. (2001).

4 Data Quality

We assessed data quality and stated semantic constraints on data, cf. Neiling et al. (2003) and Neiling (2004). These constraints determine the quality of attributes, especially regarding their identifying power. For instance, an attribute set that is stated as *approximative key* with high confidence, would be an appropriate candidate for identification.

Constraints can be stated for the attributes of single relations. Let A be a table with the attributes $Y_1, \ldots, Y_m$, $Y \subset \{Y_1, \ldots, Y_m\}$, and $a, b \in A$. $Y(a)$ denotes the value(s) of the attribute(s) Y for the tuple a, and $a \equiv b$ abbreviates that tuples a and b are matched. $dist : \mathrm{dom}(Y) \times \mathrm{dom}(Y) \to R_{\geq 0}$ denotes a distance measure on the domain of Y, and $p \in (0, 1]$.

There are two concepts for keys, which are both modified towards an approximation in order to cope with dirty data. These keys can be determined from samples of pairs. A *semantic key* is an attribute set, that identifies real-world objects in reality, but in databases it could fail, therefore we weaken it by means of conditional probabilities $\mathbf{P}(\,\cdot \mid \cdot\,)$.

- Y is an *semantic key*, if $\big(Y(a) = Y(b) \Longleftrightarrow a \equiv b\big)$
- Y is an *approximate key* with confidence p, if both

$$\begin{aligned} \text{accuracy} &:= \mathbf{P}(\, Y(a) = Y(b) \mid a \equiv b\,) \geq p, \text{ and} \\ \text{confidence} &:= \mathbf{P}(\, a \equiv b \mid Y(a) = Y(b)\,) \geq p, \end{aligned}$$

- Y is an Δ*–approximate key* with confidence p, if both

$$\begin{aligned} \Delta\text{–accuracy} &:= \mathbf{P}(\, \mathit{dist}(Y(a), Y(b)) \leq \Delta \mid a \equiv b\,) \geq p, \text{ and} \\ \Delta\text{–confidence} &:= \mathbf{P}(\, a \equiv b \mid \mathit{dist}(Y(a), Y(b)) \leq \Delta\,) \geq p, \end{aligned}$$

Differentiating keys are used to separate sets of objects: Whenever the values differ, they can not be considered to be equal. Consequently, these keys are useful for preselection, cf. section 5.

- Y is an *differentiating key*, if $\big(Y(a) \neq (b) \Longrightarrow a \not\equiv b\big)$
- Y is an *approximative differentiating key* with confidence p, if

$$\Delta\text{–anti–confidence} := \mathbf{P}(\, a \not\equiv b \mid \mathit{dist}(Y(a), Y(b)) \leq \Delta\,) \geq p,$$

Further constraints cope with the occurence of missing values, the selectivity of attributes, or the expected number of duplicates between to subsets of records, cf. Neiling et al. (2003).

5 Pair Construction/Preselection Of Pairs

To be efficient for large databases, preprocessing is applied. Obviously it is unnecessary to compare *all* pairs — most of them can be omitted. But the question arises, which pairs are to be built for comparison? Different methods exist, cf. Baxter et al. (2003). Also well-known is the so-called *Sorted Neighborhood Method*, where the records are sorted w.r.t. a combined key and pairs are built for records, that are at most k positions away w.r.t. the sorting, cf. Hernandez (1996). The choice of a preselection was described as optimization problem by Neiling and Müller (2001) and later revised by Neiling (2004).

Let δ' be a classifier for pairs of elements from two databases A_1, A_2. Within the preprocessing we avoid pairs of elements that are not likely to be matched. T.i. we use a combination $\sigma = \bigcup_j (\bigcap_i \sigma_{ij})$ of *selectors* σ_{ij}, where every σ_{ij} filters pairs from the cross product space $A_1 \times A_2$. Then we can apply the classifier $\delta = \delta' \circ \sigma$ for object identification, reducing the number of pairs to check.

The main idea behind a preselection is to employ approximative and differentiating keys efficiently. A preselection can be established on the results of the data analysis. The identified key attribute sets can be used for selectors.

Each selector σ has *processing costs*, a *selection rate*, estimating the percentage of the selected pairs from $A_1 \times A_2$:

$$\mathrm{sel}_{A_1 \times A_2}(\sigma) := \frac{|\sigma(A_1 \times A_2)|}{|A_1 \times A_2|}, \tag{1}$$

and an *error rate*, quantifying the portion of the not selected matches:

$$\mathrm{err}_{A_1 \times A_2}(\sigma) := 1 - \frac{|\{(a,b) \in \sigma(A_1 \times A_2) \mid a \equiv b\}|}{|\{(a,b) \in A_1 \times A_2 \mid a \equiv b\}|}. \tag{2}$$

Generally spoken, a good preselection combines a low error rate with a considerably high selection rate, such that the most non matched pairs fall out by default, whereby only a few matched pairs are left out. Typically, the lower the selection rate the better the performance of the whole identification task, since the main cost of object identification is determined by loading and processing of pairs. But obviously there is a trade-off between the error rate and the selection rate. Thus choosing a combined selector among a set of possible combinations of selectors becomes an optimization problem, whereby the solution can be found with greedy approaches, e.g. by means of branch–and–bound. Starting from the estimated selection and error rates of single selectors, the respective values for their combinations can be approximated with a heuristic. For instance, the selection rate of the intersection of two selectors lays between the maximum and the sum of their selection rates, such that we can choose the average as heuristic, cf. Neiling (2004), Ch.5.

Different optimization problems can be defined, e.g. to minimize the error rate under processing time constraints, or maximize the selection rate while bounding the error rate:

$$\max_{\sigma \in \Sigma} \overset{!}{=} \mathrm{sel}(\sigma) \text{ s.t. } \mathrm{err}(\sigma) \leq \kappa,$$

whereby Σ contains all combinations that can be constructed from given selectors by union and intersection like $\sigma_1 \cup (\sigma_2 \cap \sigma_3)$.

Example 1 *A relational selector σ poses conditions on attribute values, e.g. requiring equality (this is sometimes called blocking) or containment of a value in a list, or limiting the variation of cardinal scaled attributes by some $\Delta > 0$.*

Index structures, such as bitmaps or tree-based structures, are available in database management systems and can be used to achieve efficient data access.

Example 2 *A metrical selector σ poses conditions on attributes in terms of a given (multidimensional) metric dist$(\cdot,\cdot)$, e.g. the Minimum–Edit–Distance for strings. A metrical selector allows (1) the selection of the k nearest neighbors of an element, or (2) the selection of all elements within a Δ–environment for $\Delta > 0$.*

Metrical index structures can be employed, e.g. the M-tree or the MVD-tree, cf. Ciaccia et al. (1997) and Bozkaya et al. (1999). Canopy clustering *could be applied alternatively to an index, whereby a simple-to-compute 'rough' metric dist′ is used for clustering (dist′ holds for all x, y: if dist′$(x, y) \leq \xi$ then also dist$(x, y) \leq \xi$.), cf. McCallum (2000).*

Claim 1 *Let σ be a selector with approximately constant selection rate, i.e. for large sets $A_1 \times A_2$ and $A'_1 \times A'_2$, holds:*

$$sel_{A_1 \times A_2}(\sigma) \approx sel_{A'_1 \times A'_2}(\sigma). \tag{3}$$

Then its computational complexity increases quadratically with the maximal size of the databases, written $O(n^2)$.

Claim 2 *Let σ be a selector where the number of pairs to build per record is bounded by some fixed $k \in \mathbb{N}$, i.e. for any $a \in A_1$ and large sets A_2, holds:*

$$\sigma(\{a\} \times A_2) \leq k. \tag{4}$$

Then its computational complexity increases linearly with the maximal size of the databases, written $O(n)$.

The proofs of the claims can be found in Neiling (2004), Ch. 6. It follows immediately

Proposition 1 *A k-Nearest Neighbor selector has linear complexity.*

Proposition 2 *Let the domain of an attribute set Y be bounded.*[1] *Then a relational selector based on Y has quadratic complexity.*

Nevertheless, selectors with quadratic complexity are required to guarantee small error rates for large databases. It will not be sufficient to limit the number of comparisons per record, if the database size increases. Moreover, if the number of similar records exceeds such a limit, not all possible pairs will be built. For instance, if the preselection contains pairs where the last and first names equal, there might be too many records of persons named *John Smith.* In practice, the suitable number of pairs to built for a record depends on its values and should not be limited in advance.

Special attention is paid to the sampling procedure, since it is strongly related to the preselection.

[1] *Bounded* means for a continuous scaled domain, that it is bounded by an interval, while for other domains it means that the number of possible values is limited.

Sampling. The correctness of an induced classifier depends on the chosen sample it was learned from. Differently from standard learning problems, we do not have any set of instances available. Instead we have to create samples of pairs from a given database, and have to assign the labels 'match'/'non match' to them afterwards. The label assignment should be based on a reference lookup table of matched pairs, that could be either constructed manually beforehand or provided together with a benchmark data set (e.g. we got the references for the address database). We apply stratified sampling with strata for matched pairs and non matched pairs, respectively.

Parameters for sampling are the sample size N, N_1/N, the small portion of random pairs sampled from the whole cross product space, the portion of random pairs N_2/N out of the preselection, and the portion of matched pairs N_3/N that shall be contained in the sample. Obviously, $N = N_1 + N_2 + N_3$ holds. If $N_3 = 0$, the number of matched pairs is not controlled and could consequently vary (in this case it depends on the likelihood to select randomly matched pairs).

We applied stratified sampling as follows:

1. Create one stratum S_1 of random pairs of size N_1 from the whole cross product space.
2. Create one stratum S_2 of random pairs of size N_2 out of the preselection.
3. Assign the correct labels to the pairs in $S_1 \cup S_2$.
4. Determine the number n of matched pairs that are already contained in $S_1 \cup S_2$, and add n further (but only non matched) pairs out of the preselection.[2] Stop if the sample size N is reached.
5. Create a stratum S_3 by adding of $\max(0, N_3 - n)$ random pairs out of the reference set of matched pairs.

To apply supervised learning, the samples have to be split into learn- and test-samples, again with the possibility to restrain with the strata above, e.g. to require that the proportions of matches and non-matches are equal for both.

Although the sampling seems to be too complicated for our purposes, there exist no alternatives as we argue in the following.

- It is absolutely necessary to consider pairs out of the preselection for sampling, since the induced classifier will be applied to exactly such pairs afterwards. Otherwise, if the samples would be generated differently, any learned classifier will be biased. In fact, if the sample would be chosen from a superset of the preselection, decision rules voting for matches in regions outside of the preselection could not be performed. On the other hand, if the sample would be chosen from a subset of the preselection, the induced classifier would have to be applied to regions it had not been learned from, such that no prediction accuracy could be guaranteed.

[2] We can choose pairs from the preselection only, since there is nearly no chance to get a matched pair from the cross product space at random.

- The supplement with a few randomly generated pairs from the whole cross product space is appropriate, since a preselection with high selectivity excludes many negative examples, while the inclusion of some of them might lead to sharper classifiers. If only pairs with similar values are filtered, a learner might be improved with the supplemented pairs. Our experience shows, that a portion of about 5–10% works well.
- To control the portion of matched pairs is important, since the likelihood to randomly select a matched pair (even from the preselection) is usually very small. Thus, the portion of matched pairs would be small, which would be problematically for learners, that are not capable to cope adequately with skewed class distributions. Typically, the portion N_3/N is set to $\frac{1}{2}$, such that the samples will be well-balanced.

The main drawback of this sampling procedure lies in its dependency on the chosen preselection. Therefore the preselection shall cover almost all matches and thereby exclude most of the non matches. This goal can be achieved, if the identifying as well as discriminating attributes are detected by means of data quality analysis and the preselection is chosen as solution of an optimization problem as sketched above.

6 Evaluation

We selected the methods *Record Linkage*, *Decision Tree Induction*, and *Association Rule–based Classification*. The methods were tested on several samples of different sizes sampled from three databases: Address data, apartment advertisements, and bibliographic data.

Different parameters were set for classification models, e.g. the attributes (and respective comparison functions) to be taken into consideration (ranging from 4 to 14 attributes), parameters such as the pruning strategy (information gain, information gain ratio, or Gini index) and the measure to be applied by the Decision Tree Learner, the interaction model for Record Linkage, and the conflict resolution strategy for Association Rule–based Classification. We specified between 6 and 12 different classification models per method.

We present results for address data: The database consists of 250.000 records, and provides information on name, address, and birth date of German customers. We assessed the correctness of the induced classifiers on test samples by means of the *False Negative Rate*, which indicates the portion of undetected matches, and the *False Positive Rate*, estimating the misclassification rate for non matches. The scatter plot in figure 2 displays the results of the three classification models that performed best among each of the tested methods. We can state, that the Decision Tree classifier outperformed the other classifiers. It can also be seen, that for the larger samples the classifiers got more accurate. Exceptionally, *Association Rule–based Classification* did not improve with increasing sample size. Decision Trees were quite robust w.r.t. their parameterization: Regardless of the chosen measure and the

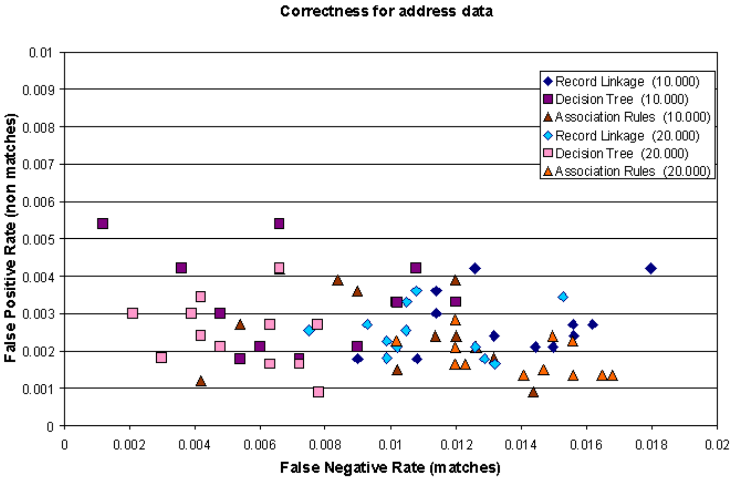

Fig. 2. Correctness results of three induced classifiers.

pruning strategy, all classifiers behaved well, with slightly better results if pruning was discarded, and the best measure was information gain ratio.

Decision Tree Induction is capable to cope with all attributes at once, while the others methods did work well only if less than 6 attributes were considered. The accuracy of the other methods depends on their parameterization: The more accurate the interaction model is specified for Record Linkage, the more correct the estimator of the multinomial distribution will be. Especially the number of attributes that were used for learning had an impact on the accuracy. Record Linkage works well for correctly specified interaction models and not too many attributes. Association Rule–based Classification does not seem to be stable enough in general, but could be used to control one of the error rates efficiently.

We conclude, that without human expertise only Decision Tree Induction yields sufficient accuracy. Unfortunately, it does not allow to control the error rates, i.e. to bound the *False Negative Rate*. This feature is required by many object identification applications. The other methods support it, since they provide a score for each pair. Record Linkage, for instance, allows to reduce the *False Negative Rate* by lowering the bound λ_l for the Likelihood Ratio (compare section 3). From the set of derived fine-grained Association Rules classifiers can be constructed, that minimize one of both error rates.

7 Summary and Outlook

We developed an universal framework for object identification. Attributes can be selected for the classification and for the preselection based on data quality analysis. Object identification is perceived as specific classification problem. Different learning methods can be applied, exemplarily we compared three methods. We discovered from our evaluation, that the use of Decision Tree Induction is well-suited for object identification. Moreover, it yielded higher accuracy and was more robust than the other methods. But it fails to control the error rates, a feature which is provided by the other investigated methods, Record Linkage and Association Rule–based Classification.

The creation of benchmark databases is a main challenge for the research community. For instance, we have made the apartment advertisements database available to other researchers.

This framework lays the foundation for future research. Other approaches could be tested based on it. For instance, which conditions have to be fulfilled, such that unsupervised learning (which does not need labelled samples at all) could be applied successfully? Or how could interactive learning (e.g. incorporation of expert suggestions and relevance feedback) or incremental learning (e.g. stepwise improvement over time) be applied?

References

ALVEY, W. and JAMERSON, B. (Eds.) (1997): *Record Linkage Techniques — 1997.* Int. Workshop, Arlington, Virginia, 1997.

BELL, G. B. and SETHI, A. (2001): *Matching records in a national medical patient index.* Communications of the ACM 44(9), 83–88.

BERTHOLD, M. and HAND, D. J. (Eds.) (1999): *Intelligent Data Analysis: An Introduction.* New York: Springer.

BILENKO, M. and MOONEY, R. (2003): Adaptive duplicate detection using learnable string similarity measures. *KDD Conf. 2003, Washington DC.*

BITTON, D. and DEWITT,D. J. (1983): *Duplicate record elimination in large data files.* ACM TODS 8(2), 255–265.

BOZKAYA, T. and ÖZSOYOGLU, Z. M. (1999): Indexing large metric spaces for similarity search queries. *ACM TODS 24*(3), 361–404.

BREIMAN, L., FRIEDMAN, J., OLSHEN R., and STONE C. (1984): *Classification and regression trees.* Chapman & Hall.

CIACCIA, P., M. PATELLA, and P. ZEZULA (1997). M-tree: An efficient access method for similarity search in metric spaces. *VLDB 1997*, pp. 426–435.

ELFEKY, M.G., VERYKIOS, V.S., and ELMAGARMID, A.K. (2002): Tailor: A record linkage toolbox. *ICDE 2002, San Jose.*

FELLEGI, I. P. and SUNTER, A. B. (1969): *A theory of record linkage.* Journal of the American Statistical Association, 64, 1183–1210.

GALHARDAS, H., FLORESCU, D., SHASHA, D., SIMON, E. and SAITA, C.-A. (2001): Declarative data cleaning: Language, model and algorithms. VLDB 2001.

GU, L., BAXTER, R., VICKERS, D., and RAINSFORD, C. (2003): Record Linkage: Current practice and future directions. Technical Report 03/83, CSIRO Mathematical and Information Sciences, Canberra, Australia.

HERNANDEZ, M. A. (1996): *A Generalization of Band Joins and The Merge/Purge Problem.* Phd thesis, Columbia University.

HERNANDEZ, M.A. and STOLFO, S.J. (1995): The merge/purge problem for large databases. *ACM SIGMOD Conf. 1995, 127-138.*

JARO, M. A. (1989): Advances in record-linkage methodology as applied to matching the census of Tampa, Florida. *JASA 84*(406), 414–420.

LIM, E.-P., SRIVASTAVA, J., PRABHAKAR, S., and RICHARDSON, J. (1993): Entity Identification in Database Integration. *ICDE 1993*, pp. 294–301.

LIU, C. and RUBIN, D. B. (1994): The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika 81*(4), 633–48.

MCCALLUM, A., NIGAM, K., and UNGAR, L. H. (2000): Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD 2000: New York, USA*, pp. 169–178.

MENG, X.-L. and RUBIN, D. B. (1993): *Maximum likelihood estimation via the* ECM *algorithm: A general framework.* Biometrika 80(2), 267–78.

MICHIE, D., SPIEGELHALTER, D. J., and TAYLOR, C. C. (1994): *Machine learning, neural and statistical classification.* New York: Horwood.

NEILING, M. (2004): *Identifizierung von Realwelt-Objekten in multiplen Datenbanken.* Dissertation, Techn. Universität Cottbus, 2004.

NEILING, M., JURK, S., LENZ, H.-J., and NAUMANN, F.: Object identification quality. *Workshop on Data Quality in Coop. Information Systems, Siena, 2003.*

NEILING, M. and JURK, S. (2003): The Object Identification Framework. *Workshop on Data Cleaning, Record Linkage and Object Consolidation at the KDD 2003, Washington DC.*

NEILING, M. and LENZ, H.-J. (2000): Data integration by means of object identification in information systems. *ECIS 2000, Vienna, Austria.*

NEILING, M. and LENZ, H.-J. (2004): *The German Administrative Record Census — An Object Identification Problem.* Allg. Stat. Arch. 88, 259–277.

NEILING, M. and R. MÜLLER (2001): The good into the pot, the bad into the crop. preselection of record pairs for database integration. *Workshop DBFusion 2001, Gommern, Germany.*

NEWCOMBE, H. B., KENNEDY, J. M., AXFORD, S. J., and JAMES, A. P. (1959): *Automatic linkage of vital records.* Science 130, 954–959.

CHRISTEN, P., CHURCHES, T., and HEGLAND, M. (2004): Febrl — a parallel open source data linkage system. *PAKDD*, Vol. 3056 of *LNCS*, pp. 638–647.

BAXTER, R., CHRISTEN, P., and CHURCHES, T. (2003): A comparison of fast blocking methods for record linkage. *Workshop on Data Cleaning, Record Linkage and Object Consolidation at the KDD 2003, Washington DC* .

TEJADA, S., KNOBLOCK, C. A., and MINTON, S. (2001): Learning object identification rules for information integration. *Information Systems 26*(8).

VERYKIOS, V., ELMAGARMID, A. and HOUSTIS, E. (2000): Automating the approximate record matching process. *J. Information Sciences 126*, 83–98.

WANG, Y. R. and MADNICK, S. E. (1989): The inter-database instance identification problem in integrating autonomous systems. *ICDE 1989*, pp. 46–55.

WINKLER, W. E. (1993): *Improved decision rules in the Fellegi-Sunter model of record linkage.* The Research Report Series, U.S. Bureau of the Census.

WINKLER, W. E. (1999): The state of record linkage and current research problems. Statistical research report series, U.S. Bureau of the Census, Washington D.C.

WINKLER, W. E. (2001): Record linkage software and methods for merging administrative lists. Statistical research report series, U.S. Bureau of the Census.

YANCEY, W. (2002): Improving parameter estimates for record linkage parameters. *Section on Survey Research Methodology. American Statistical Association.*