Evaluating Indeterministic Duplicate Detection Results

Fabian Panse and Norbert Ritter

University of Hamburg, Vogt-Kölln Straße 30, 22527 Hamburg, Germany
{panse,ritter}@informatik.uni-hamburg.de
http://vsis-www.informatik.uni-hamburg.de/

Abstract. Duplicate detection is an important process for cleaning or integrating data. Since real-life data is often polluted, detecting duplicates usually comes along with uncertainty. To handle duplicate uncertainty in an appropriate way, indeterministic duplicate detection approaches, i.e. approaches in which ambiguous duplicate decisions are probabilistically modeled in the resultant data, have been developed. To rate the goodness of a duplicate detection approach, its detection results need to be evaluated in their quality. In this paper, we propose several semantics to apply traditional quality evaluation measures to indeterministic duplicate detection results and exemplarily present an efficient evaluation for one of these semantics. Finally, we present some experimental results.

Keywords: indeterministic duplicate detection, probabilistic duplicate detection, quality evaluation, probabilistic clustering, entity resolution.

1 Introduction

Duplicate detection [4,8] is an important task in cleaning a single data source or in meaningfully combining data from different sources. Due to deficiencies like missing data, typos or data obsolescence, it often cannot be determined with absolute certainty from the data itself that two or more representations belong to the same real-world entity. This principally hinders duplicate detection and is a crucial source of uncertainty. Most current duplicate detection approaches [4] acknowledge many kinds of uncertainty and often apply fuzzy matching techniques, but in the end they still are deterministic: finally an absolute decision needs to be taken either by (1) deferring the situation to domain experts which is expensive and time consuming, or (2) choose the most likely configuration thereby risking a wrong choice with all consequences this may have.

To better deal with uncertainty in duplicate detection, several approaches [2,6,9] have been proposed that avoid ambiguous decisions, but instead try to model all significantly likely configurations in the resultant data. Hence any query answer or other derived data will reflect the inherent uncertainty. Since in such approaches duplicate decisions are handled in an indeterministic way, we refer to them as an indeterministic duplicate detection. This concept may protect against negative impact resulting from false duplicate decisions made under ambiguous circumstances.

For effectively comparing deterministic- and indeterministic duplicate detection approaches new methods for quality evaluation are required, because existing evaluation

E. Hüllermeier et al. (Eds.): SUM 2012, LNAI 7520, pp. 433–446, 2012.

measures are not designed to deal with indeterministic results. As we think, the quality of an indeterministic duplicate detection result generally depends on the intended handling of the datas' inherent uncertainty. For that reason, in this paper we define different semantics for evaluating the quality of indeterministic duplicate detection results and propose strategies to compute these evaluations in an efficient way.

The paper is structured as follows. In Section 2, we formally introduce the concepts of deterministic duplicate detection and indeterministic duplicate detection. Moreover, we present measures for evaluating the quality of deterministic duplicate detection results. In Section 3, we introduce different semantics on how the quality of an indeterministic duplicate detection result can be scored and exemplarily discuss one of them in detail. In Section 4, we present an efficient quality computation for this semantics. Section 5 shows some experimental results. In Section 6 we present related work. Finally, Section 7 concludes the work.

2 Duplicate Detection

Duplicate detection [8,4] is the process of identifying multiple representations in a database relation referring to the same real-world entity.

Definition 1 (Real World): *We postulate a real world, denoted by $\mathfrak{W}$, as the set of all existing real-world entities. The mapping $\omega : \mathcal{R} \rightarrow \mathfrak{W}$ maps tuples of a database relation $\mathcal{R}$ on entities of $\mathfrak{W}$.*

In our linguistic use, two tuples $t_1, t_2 \in \mathcal{R}$ are called duplicates, iff $\omega(t_1) = \omega(t_2)$.

2.1 Deterministic Duplicate Detection

Deterministic duplicate detection is a partitioning of the input relation into clusters (equivalence classes or partition classes) such that all tuples of one cluster refer to the same real-world entity and hence are duplicates.

Definition 2 (Deterministic Duplicate Detection): *Deterministic duplicate detection is a function δ_{det} that maps a relation $\mathcal{R}$ to a clustering $\mathcal{C} = \{C_1, \ldots, C_l\}$ such that $\bigcup \mathcal{C} = \mathcal{R}$ (each tuple is assigned to a cluster) and $(\forall C_1, C_2 \in \mathcal{C}) : C_1 \cap C_2 = \emptyset$ (the clusters are disjoint). The duplicate detection is considered to be* perfect*, iff:*

- $(\forall C \in \mathcal{C}\ \forall t_1, t_2 \in C) : \omega(t_1) = \omega(t_2)$ *, i.e., all tuples of one cluster represent the same real-world entity (the duplicate detection is correct $\Rightarrow$* precision=*1)*
- $(\forall C_1, C_2 \in \mathcal{C}\ \forall t_1 \in C_1\ \forall t_2 \in C_2) : C_1 \neq C_2 \Rightarrow \omega(t_1) \neq \omega(t_2)$*, i.e., all tuples of different clusters represent different real-world entities (the duplicate detection is complete $\Rightarrow$* recall=*1)*

To evaluate the quality of a duplicate detection process performed on $\mathcal{R}$, its resultant clustering $\mathcal{C}$ is compared with the clustering $\mathcal{C}_{\text{gold}}$ which would result from a perfect duplicate detection process (called the gold standard) on $\mathcal{R}$.

As a running example throughout this paper, we consider a duplicate detection on a relation $\mathcal{R}_{\text{Ex}}$ with the ten tuples $t_1, \ldots, t_{10}$. Figure 1 presents the gold standard and a certain clustering resultant from a non-perfect deterministic duplicate detection process.

Fig. 1. The gold standard and a non-perfect deterministic clustering result on $\mathcal{R}_{\text{Ex}}$

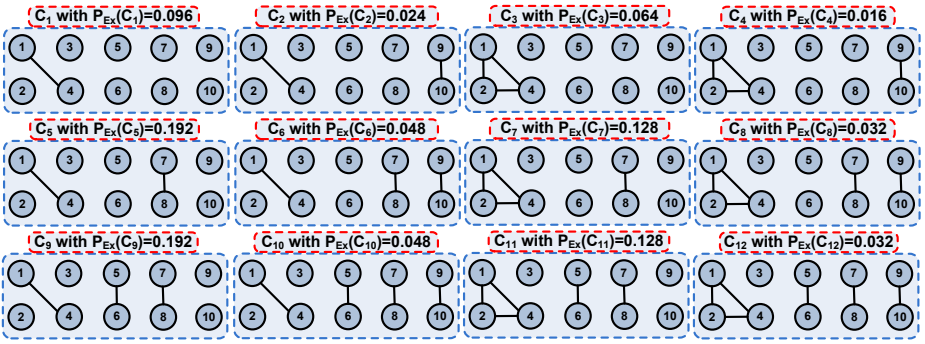


Fig. 2. The possible clusterings $\mathcal{C}_i \in \Gamma_{\text{Ex}}$ of our sample probabilistic clustering $\mathfrak{C}_{\text{Ex}} = (\Gamma_{\text{Ex}}, P_{\text{Ex}})$

2.2 Indeterministic Duplicate Detection

In contrast to a deterministic duplicate detection approach where two tuples have to be declared as duplicates or not, in an indeterministic approach duplicate decisions can be made in a probabilistic way, i.e. tuples can be declared as duplicates with a given probability. For example, the two tuples t_1 and t_2 can be declared to be duplicates with a probability of 60% (and hence to be non-duplicates with a probability of 40%).

The result of an indeterministic duplicate detection is a probability distribution on a set of possible clusterings where each clustering corresponds to a deterministic duplicate detection result.

Definition 3 (Indeterministic Duplicate Detection): *Indeterministic duplicate detection is a function δ_{idet} that maps a relation $\mathcal{R}$ to a probabilistic clustering $\mathfrak{C}=(\Gamma, P)$ where:*

- *Γ is a set of possible clusterings so that $(\forall \mathcal{C} \in \Gamma) : (\exists \delta_{det}) : \mathcal{C} = \delta_{det}(\mathcal{R})$,*
- *$P : \Gamma \rightarrow (0, 1], \sum_{\mathcal{C} \in \Gamma} P(\mathcal{C}) = 1$ is a probability distribution on Γ*

A sample probabilistic clustering $\mathfrak{C}_{\text{Ex}} = (\Gamma_{\text{Ex}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{12}\}, P_{\text{Ex}})$ of our sample input relation $\mathcal{R}_{\text{Ex}} = \{t_1, \ldots, t_{10}\}$ is graphically presented in Figure 2.

Definition 4 (Cross Product of Probabilistic Clusterings): *The cross product of two probabilistic clusterings $\mathfrak{C}_i = (\Gamma_i, P_i)$ and $\mathfrak{C}_j = (\Gamma_j, P_j)$ is the probabilistic clustering $\mathfrak{C}_{ij} = \mathfrak{C}_i \times \mathfrak{C}_j = (\Gamma_{ij}, P_{ij})$ where $\Gamma_{ij} = \{\mathcal{C}_i \cup \mathcal{C}_j \mid \mathcal{C}_i \in \Gamma_i, \mathcal{C}_j \in \Gamma_j\}$ and the probability of each resultant possible clustering $\mathcal{C} = \mathcal{C}_i \cup \mathcal{C}_j$ is $P_{ij}(\mathcal{C}) = P_i(\mathcal{C}_i) \cdot P_j(\mathcal{C}_j)$.*

The n-ary cross product is defined accordingly.

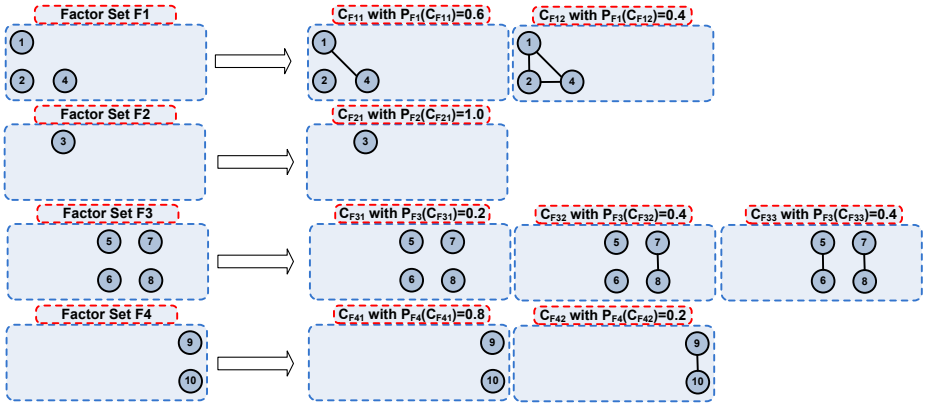


Fig. 3. The four factors of $\mathfrak{C}_{\text{Ex}}$: $\mathfrak{C}_{F1}$, $\mathfrak{C}_{F2}$, $\mathfrak{C}_{F3}$ and $\mathfrak{C}_{F4}$

Because the data of individual clusterings often considerably overlaps and it is sometimes even impossible to store them separately (in our experiments we work on probabilistic clusterings with $|\Gamma| \rightarrow 7.6 \cdot 10^{44}$), a succinct representation has to be used. For that reason, a probabilistic clustering is usually represented in a factorized way.

Definition 5 (Factorization of a Probabilistic Clustering): A factorization of a probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ defined on a relation $\mathcal{R}$ is a set of probabilistic clusterings (called factors) $\mathcal{F}(\mathfrak{C}) = \{\mathfrak{C}_{F1}, \ldots, \mathfrak{C}_{Fn}\}$ where each factor is defined on a tuple set $Fi \subset \mathcal{R}$ (called factor set) so that the following three requirements are satisfied:

- Each tuple $t \in \mathcal{R}$ is covered by a factor (the factorization is lossless): $\bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} F$
- The overall probabilities of the individual clusterings are preserved (the factorization is probability correct): $(\forall \mathfrak{C}_F \in \mathcal{F}(\mathfrak{C}) \; \forall \mathcal{C}_F \in \mathfrak{C}_F) : P_F(\mathcal{C}_F) = \sum_{\mathcal{C} \in \Gamma, \mathcal{C}_F \subseteq \mathcal{C}} P(\mathcal{C})$
- Each two factors $\mathfrak{C}_{Fi}$ and $\mathfrak{C}_{Fj}$ are independent to each other (the factorization is correct): $(\forall \mathcal{C}_1 \in \mathfrak{C}_{Fi} \; \forall \mathcal{C}_2 \in \mathfrak{C}_{Fj}) : P_{Fi}(\mathcal{C}_1) \cdot P_{Fj}(\mathcal{C}_2) = \sum_{\mathcal{C} \in \Gamma, \mathcal{C}_1 \cup \mathcal{C}_2 \subseteq \mathcal{C}} P(\mathcal{C})$. This implies that each two factors $\mathfrak{C}_{Fi}$ and $\mathfrak{C}_{Fj}$ are defined on disjoint factor sets, i.e. $F_i \cap F_j = \emptyset$.

The factorization is complete, iff none of its factors can be further factorized. Due to a factorization is correct and lossless, Theorem 1 is valid:

Theorem 1. *A probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ can be rebuilt from the cross product of its factors: $\mathfrak{C} = \times_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} \mathfrak{C}_F$*

Proof. The proof directly results from the definition of the cross product and the definition of a correct and lossless factorization.

Due to the number of possible clusterings is usually overwhelming, existent approaches of indeterministic duplicate detection [2,6,9] are designed in a way that they already produce a factorized representation as output.

Figure 3 shows the four factors of our sample probabilistic clustering $\mathfrak{C}_{\text{Ex}} = (\Gamma_{\text{Ex}}, P_{\text{Ex}})$ along with their factor sets and their sets of possible clusterings.

2.3 Quality Evaluation Measures

Existing quality measures for deterministic duplicate detection [5,7,11] can be classified into decision-based evaluation measures and cluster-based evaluation measures.

Decision-Based Evaluation. Traditional approaches for duplicate detection [8] are based on pairwise tuple comparisons. For that reason, the quality of a duplicate detection approach is often measured based on the pairwise duplicate decisions made by this approach. The two most known decision-based evaluation measures are recall and precision [11] which originate from the area of information retrieval.

Before a decision-based evaluation measures can be applied to a clustering, the clustering needs to be transformed into a set of pairwise decisions. A transformation from a clustering $\mathcal{C}$ to the corresponding set of duplicate decisions[1] (the set of proposed matches M and the set of proposed unmatches U) can be defined as:

$$M(\mathcal{C}) = \{(t_i, t_j) \mid t_i, t_j \in \mathcal{R} \wedge (\exists C \in \mathcal{C}) : \{t_i, t_j\} \subseteq C\} \tag{1}$$

$$U(\mathcal{C}) = \{(t_i, t_j) \mid t_i, t_j \in \mathcal{R} \wedge (\nexists C \in \mathcal{C}) : \{t_i, t_j\} \subseteq C\} \tag{2}$$

From these two sets three decision classes, the true positives (TP), the false positives (FP), and the false negatives (FN) can be derived as:

$$\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}}) = M(\mathcal{C}) \cap M(\mathcal{C}_{\text{gold}}) \tag{3}$$

$$\text{FP}(\mathcal{C}, \mathcal{C}_{\text{gold}}) = M(\mathcal{C}) \cap U(\mathcal{C}_{\text{gold}}) = M(\mathcal{C}) - M(\mathcal{C}_{\text{gold}}) \tag{4}$$

$$\text{FN}(\mathcal{C}, \mathcal{C}_{\text{gold}}) = U(\mathcal{C}) \cap M(\mathcal{C}_{\text{gold}}) = M(\mathcal{C}_{\text{gold}}) - M(\mathcal{C}) \tag{5}$$

Using these three classes, recall (Rec) and precision (Prec) can be defined as:

$$\text{Rec}(\mathcal{C}, \mathcal{C}_{\text{gold}}) = \frac{|\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})|}{|M(\mathcal{C}_{\text{gold}})|} = \frac{|\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})|}{|\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})| + |\text{FN}(\mathcal{C}, \mathcal{C}_{\text{gold}})|} \tag{6}$$

$$\text{Prec}(\mathcal{C}, \mathcal{C}_{\text{gold}}) = \frac{|\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})|}{|M(\mathcal{C})|} = \frac{|\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})|}{|\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})| + |\text{FP}(\mathcal{C}, \mathcal{C}_{\text{gold}})|} \tag{7}$$

A third measure that combines precision and recall into a single quality score by computing their harmonic mean is the F_1-score:

$$F_1\text{-score}(\mathcal{C}, \mathcal{C}_{\text{gold}}) = 2 \cdot \frac{\text{Rec}(\mathcal{C}, \mathcal{C}_{\text{gold}}) \cdot \text{Prec}(\mathcal{C}, \mathcal{C}_{\text{gold}})}{\text{Rec}(\mathcal{C}, \mathcal{C}_{\text{gold}}) + \text{Prec}(\mathcal{C}, \mathcal{C}_{\text{gold}})} \tag{8}$$

$$= \frac{2 \cdot |\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})|}{2 \cdot |\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})| + |\text{FP}(\mathcal{C}, \mathcal{C}_{\text{gold}})| + |\text{FN}(\mathcal{C}, \mathcal{C}_{\text{gold}})|} \tag{9}$$

Further decision-based evaluation measures are proposed in [5,7]. When clear from context, we often simply use TP, FP, FN, Rec, Prec and F_1-score instead of $\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}})$, $\text{FP}(\mathcal{C}, \mathcal{C}_{\text{gold}})$, $\text{FN}(\mathcal{C}, \mathcal{C}_{\text{gold}})$, $\text{Rec}(\mathcal{C}, \mathcal{C}_{\text{gold}})$, $\text{Prec}(\mathcal{C}, \mathcal{C}_{\text{gold}})$ and $F_1\text{-score}(\mathcal{C}, \mathcal{C}_{\text{gold}})$.

[1] Note, most often only the positive duplicate decisions need to be computed.

Cluster-Based Evaluation. In measures for cluster-based evaluation the quality of a duplicate detection process is scored by the similarity of its final clustering to the perfect clustering. The more similar both clusterings (partitions) are, the better is the process's quality. The most of these approaches [11], e.g. the Rand Index, the Adjusted Rand Index, and the Talburt-Wang Index, are based on the *partition overlap* of the two clusterings to be compared. According to [11], the partition overlap V of two partitions $\mathcal{C}_A$ and $\mathcal{C}_B$ is the set of all nonempty intersections between the clusters of $\mathcal{C}_A$ and the clusters of $\mathcal{C}_B$ and is defined as:

$$V(\mathcal{C}_A, \mathcal{C}_B) = \{A_i \cap B_j \mid A_i \in \mathcal{C}_A, B_j \in \mathcal{C}_B \wedge A_i \cap B_j \neq \emptyset\} \tag{10}$$

Whereas the Rand Index and the Adjusted Rand Index are computationally intensive, the Talburt-Wang Index (short TWI) is simply to calculate, because it does not use the size of the overlaps, but only the number of overlaps:

$$\text{TWI}(\mathcal{C}_A, \mathcal{C}_B) = \frac{\sqrt{|\mathcal{C}_A| \cdot |\mathcal{C}_B|}}{|V(\mathcal{C}_A, \mathcal{C}_B)|} \tag{11}$$

In this paper, we will use the TWI as a representative for cluster-based evaluation measures. Table 1 depicts the quality scores of the twelve possible clusterings of our sample probabilistic clustering $\mathfrak{C}_{\text{Ex}}$ w.r.t. the four presented evaluation measures.

Table 1. Quality scores of the possible clusterings of $\mathfrak{C}_{\text{Ex}}$

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_6$	$\mathcal{C}_7$	$\mathcal{C}_8$	$\mathcal{C}_9$	$\mathcal{C}_{10}$	$\mathcal{C}_{11}$	$\mathcal{C}_{12}$	**min**	**max**	**exp**
Rec	.333	.333	.333	.333	.667	.667	.667	.667	.667	.667	.667	.667	.333	.670	.600
Prec	1.00	.500	.333	.250	1.00	.667	.500	.400	.667	.500	.400	.333	.250	1.00	.648
F_1**-score**	.500	.400	.333	.286	.800	.667	.571	.500	.667	.571	.500	.444	.286	.800	.592
TWI	.882	.881	.831	.778	.935	.875	.875	.810	.875	.810	.810	.740	.740	.935	.864

3 Quality of Indeterministic Duplicate Detection Results

In this section, we analyze the different types of on-top applications which can process indeterministic duplicate detection results and define specific quality semantics for each of them (Section 3.1). For exemplary reasons, we go then into detail with one of these semantics in Section 3.2. A closer consideration of the other semantics is intended for future publications.

3.1 Quality Semantics

The quality of data generally depends on its intended use, i.e. a database can be of good quality w.r.t. a given application and can be of bad quality w.r.t. another one. The same holds for the quality of a duplicate detection process, because its goodness is automatically a quality yardstick of the data resulting from deduplication, i.e. the more error the duplicate detection process produces, the worse is the quality of the resultant data.

We identify four different ways of handling uncertainty in data processing and hence classify four different types of on-top applications (see Figure 4):

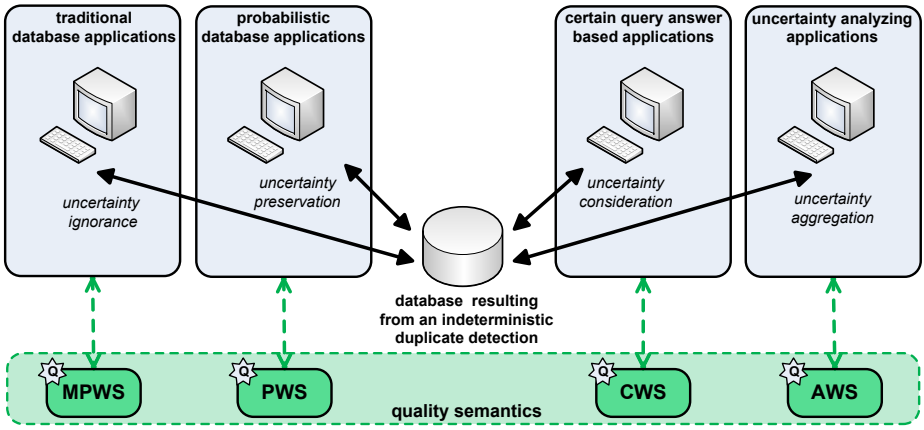
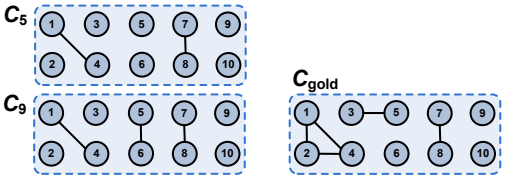


Fig. 4. The four types of database applications along with their corresponding quality semantics

1. **Traditional Database Applications (Uncertainty Ignorance):** Most of traditional database applications cannot process probabilistic data and need certain data as input. In this case, uncertainty must be ignored by evaluating queries only on one of the possible worlds (most meaningful: one of the most probable worlds).
2. **Probabilistic Database Applications (Uncertainty Preservation):** Another way an application can handle data uncertainty is to consider any kind of uncertainty during query evaluation and to produce an uncertain query result. Evaluating a query on a probabilistic database follows the principles of the possible world semantics [10]. This means that the query is evaluated in each world individually and each result represents a possible world of the probabilistic query answer.
3. **Certain Query Answer based Applications (Uncertainty Consideration):** A lot of applications require query answers as input, which are (nearly) dead certain. For that reason *consistent query answering* [1] (also known as *certain query answering* or *sure information answering*) need to be applied to the uncertain data. In this case, uncertainty is resolved by processing a query only on the certain facts, or at least on the facts which are certain with a given level of tolerance, of the probabilistic database. It is important to note that indeterministic duplicate detection allows a more correct evaluation of certain query answering than it is possible by querying a deterministic duplicate detection result, because the query answering algorithm can distinct between ambiguous duplicate decisions and certain duplicate decisions.
4. **Uncertainty Analyzing Applications (Uncertainty Aggregation):** The last type includes applications which are designed to directly analyze the uncertainty of the data, as for example to compute the minimal/maximal/expected number of database tuples which satisfy a specific selection criterion (e.g.: *What is the minimal/maximal/expected number of persons living in Germany*). In this case, aggregation functions are used to resolve data uncertainty.

Since the quality of an indeterministic duplicate detection result essentially depends on the way the resultant datas' inherent uncertainty is processed by an on-top application, we define the following four corresponding quality semantics:

1. **Most Probable World Semantics (short MPWS):** The most probable world semantics is designed to score the quality of an indeterministic duplicate detection result w.r.t. traditional database applications. If the application picks one of the most probable clusterings (worlds) randomly, it is most meaningful to score the final quality as the average quality of the most probable clusterings. Of course, if any other selection criterion is used, another quality definition can be more meaningful. For our sample probabilistic clustering $\mathfrak{C}_{\text{Ex}}$ the two possible clusterings $\mathcal{C}_5$ and $\mathcal{C}_9$ are most probable. These clusterings along with the final quality scores are presented in Figure 5.



(a) Most Probable Clusterings (b) Gold Standard

	$\mathcal{C}_5$	$\mathcal{C}_9$	$\varnothing$
Recall	0.667	0.667	0.667
Precision	1.0	0.667	0.8335
F_1-score	0.8	0.667	0.73
TWI	0.935	0.875	0.91

(c) Quality Scores

Fig. 5. $\mathfrak{C}_{\text{Ex}}$: its most probable clusterings (a), its gold standard (b) and the quality scores (c)

2. **Possible World Semantics (short PWS):** If the data is processed according to the possible world semantics, it seems most meaningful to define the quality of an indeterministic duplicate detection result as a probability distribution on all possible scores (the possible worlds of the datas' quality). Moreover, this approach allows a subtle analysis of the probabilistic clustering's quality. For example, it allows to determine to what probability is the quality score greater than a user specific threshold. Nevertheless, this semantics is computationally intensive.
For our sample probabilistic clustering $\mathfrak{C}_{\text{Ex}}$, the resultant probability distribution on possible F_1-scores is presented in Figure 6.

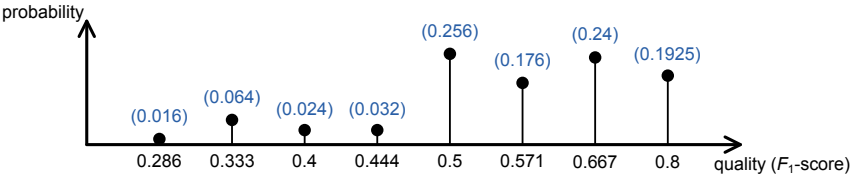


Fig. 6. Probability distribution on possible F_1-scores of $\mathfrak{C}_{\text{Ex}}$

3. **Certain World Semantics (short CWS):** The certain world semantics is tailor-made for applications based on certain query answers. Thus, in the CWS the quality of the duplicate detection result is only scored on clustering information which is postulated to be certain. The CWS is extensively discussed in Section 3.2.

4. **Aggregated World Semantics (short AWS):** For uncertainty analyzing applications, it is most meaningful to score the final quality of an indeterministic duplicate detection result by the quality of the analysis result and hence by aggregating (e.g. by $\min$, $\max$ or $\exp$) the quality scores of all its possible clusterings. This semantics

allows a rough analysis of the datas' quality. For example it allows to query: *What is the worst case scenario (minimal quality score), the expected scenario (expected quality score), and the best case scenario (maximal quality score)*. The aggregated quality scores of $\mathfrak{C}_{\text{Ex}}$ are listed in Table 1.

It is important to note that these semantics are no competitors in general, but each of them fits best for a specific application scenario.

3.2 Certain World Semantics

In the certain world semantics we evaluate the quality only on the facts postulated to be dead certain (or more probable than $1-\epsilon$ respectively). We have to differentiate between a cluster-based interpretation and a decision-based interpretation. Whereas the cluster-based interpretation considers only clusters with a certain existence, the decision-based interpretation considers certain duplicate decisions.

Cluster-Based Interpretation. Intuitively, a cluster C with $|C| > 1$ can be considered to be certain, if in each possible clustering there exists a cluster C' with $C \subseteq C'$. However, under a closer consideration, we will see that this intuitive definition is not appropriate, because a certain cluster $\{t_1, t_2, t_3\}$ not only means that t_1, t_2 and t_3 are certainly duplicates, but it also implicitly means that these three tuples are certainly no duplicates with any other tuple what in reality muss not be a certain fact at all.

As a consequence, we consider the certain clustering component $C_{\text{cert.}\epsilon}(\mathfrak{C})$ of a probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ to be the set of clusters which belong to every possible clustering of $\mathfrak{C}$ with a probability equal to 1-ϵ or greater.

Definition 6 (Certain Clustering Component): *Let $\mathfrak{C} = (\Gamma, P)$ be a probabilistic clustering. The certain clustering component of $\mathfrak{C}$ with the tolerance setting ϵ is a traditional clustering defined as:*

$$C_{cert.\text{-}\epsilon}(\mathfrak{C}) = \{C \mid \sum\nolimits_{\mathcal{C} \in \Gamma, C \in \mathcal{C}} P(\mathcal{C}) \geq 1 - \epsilon\} \tag{12}$$

By definition, the certain clustering component of a probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ can be contain less tuples than $\mathcal{R}$ and hence less tuples than the gold standard. To enable a meaningful execution of a cluster-based evaluation measure as the Talburt-Wang index, we have to modify $\mathcal{C}_{\text{gold}}$ so that it shares the same tuples as $C_{\text{cert.}\text{-}\epsilon}(\mathfrak{C})$. For that purpose, we discard all clusters from $\mathcal{C}_{\text{gold}}$ which do not have a tuple belonging to $C_{\text{cert.}\text{-}\epsilon}(\mathfrak{C})$ and then drop from the remaining clusters all tuples which do not belong to any cluster of $C_{\text{cert.}\text{-}\epsilon}(\mathfrak{C})$. Formally, the modifications are defined as:

$$T_{\text{cert.}\text{-}\epsilon}(\mathfrak{C}) = \bigcup \mathcal{C}_{\text{cert.}\text{-}\epsilon}(\mathfrak{C}) = \{t \mid (\exists C \in \mathcal{C}_{\text{cert.}\text{-}\epsilon}(\mathfrak{C})) : t \in C\} \tag{13}$$

$$\mathcal{C}^*_{\text{gold}} = \{C \cap T_{\text{cert.}\text{-}\epsilon}(\mathfrak{C}) \mid C \in \mathcal{C}_{\text{gold}}\} - \emptyset \tag{14}$$

Let q be the cluster-based quality measure to score, the quality of the probabilistic clustering $\mathfrak{C}$ to $\mathcal{C}_{\text{gold}}$ using q under the CWS with the tolerance setting ϵ is scored as:

$$q(\mathfrak{C}, \mathcal{C}_{\text{gold}})_{\text{CWS},\epsilon} = q(\mathcal{C}_{\text{cert.}\text{-}\epsilon}(\mathfrak{C}), \mathcal{C}^*_{\text{gold}}) \tag{15}$$

As an example consider Figure 7. The certain clustering component of $\mathfrak{C}_{\text{Ex}}$ with the tolerance setting $\epsilon = 0.3$ is $\mathcal{C}_{\text{cert.-}0.3}(\mathfrak{C}) = \{\{t_3\}, \{t_7, t_8\}, \{t_9\}, \{t_{10}\}\}$. The modified gold standard is $\mathcal{C}^*_{\text{gold}} = \{\{t_3\}, \{t_7, t_8\}, \{t_9\}, \{t_{10}\}\}$. Thus, the F_1-score as well as the TWI of $\mathfrak{C}_{\text{Ex}}$ are 1.0. In contrast, by using the tolerance setting $\epsilon = 0.4$ the certain clustering component is equivalent to $\mathcal{C}_9$ and the gold standard remained unchanged. Thus, the resultant scores of $\mathfrak{C}_{\text{Ex}}$ are 0.667 (F_1-score) and 0.875 (TWI).

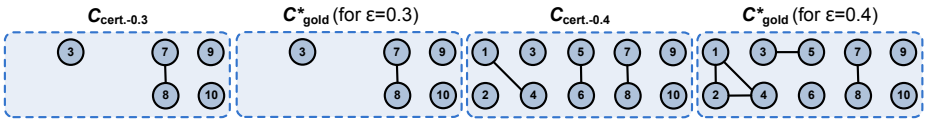


Fig. 7. $\mathcal{C}_{\text{cert.-}\epsilon}(\mathfrak{C})$ and $\mathcal{C}^*_{\text{gold}}$ for the tolerance settings $\epsilon = 0.3$ and $\epsilon = 0.4$

Decision-Based Interpretation. The set of certain decisions of a probabilistic clustering $\mathfrak{C}$ is the set of decisions which are postulated with a probability greater than $1 - \epsilon$.

Definition 7 (Certain Duplicate Decisions): *Let $\mathfrak{C} = (\Gamma, P)$ be a probabilistic clustering. The set of certain positive (negative) duplicate decisions of $\mathfrak{C}$ with the tolerance setting ϵ are defined as:*

$$M_{cert.\text{-}\epsilon}(\mathfrak{C}) = \{(t_i, t_j) \mid t_i, t_j \in \mathcal{R} \wedge \sum\nolimits_{\mathcal{C} \in \Gamma, (t_i,t_j) \in M(\mathcal{C})} P(\mathcal{C}) \geq 1 - \epsilon\} \quad (16)$$

$$U_{cert.\text{-}\epsilon}(\mathfrak{C}) = \{(t_i, t_j) \mid t_i, t_j \in \mathcal{R} \wedge \sum\nolimits_{\mathcal{C} \in \Gamma, (t_i,t_j) \in U(\mathcal{C})} P(\mathcal{C}) \geq 1 - \epsilon\} \quad (17)$$

$$= \{(t_i, t_j) \mid t_i, t_j \in \mathcal{R} \wedge \sum\nolimits_{\mathcal{C} \in \Gamma, (t_i,t_j) \in M(\mathcal{C})} P(\mathcal{C}) \leq \epsilon\} \quad (18)$$

The three decision classes TP, FP, and FN can be then computed as follows:

$$\text{TP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) = M_{\text{gold}} \cap M_{\text{cert.-}\epsilon}(\mathfrak{C}) \quad (19)$$

$$\text{FP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) = M_{\text{cert.-}\epsilon}(\mathfrak{C}) - M_{\text{gold}} \quad (20)$$

$$\text{FN}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) = M_{\text{gold}} \cap U_{\text{cert.-}\epsilon}(\mathfrak{C}) \quad (21)$$

Recall, precision and F_1-score are then computed according to Equations 6-9 by using $\text{TP}_{\text{cert.-}\epsilon}$, $\text{FP}_{\text{cert.-}\epsilon}$, and $\text{FN}_{\text{cert.-}\epsilon}$ instead of TP, FP, and FN.

Let q be the decision-based quality measure to score, let m be the evaluation method performed in q having the three decision classes TP, FN, and FP as input: $q(\mathcal{C}, \mathcal{C}_{\text{gold}}) = m(\text{TP}(\mathcal{C}, \mathcal{C}_{\text{gold}}), \text{FP}(\mathcal{C}, \mathcal{C}_{\text{gold}}), \text{FN}(\mathcal{C}, \mathcal{C}_{\text{gold}}))$. The quality of $\mathfrak{C}$ to $\mathcal{C}_{\text{gold}}$ using q under the CWS with the tolerance setting ϵ is scored as:

$$q(\mathfrak{C}, \mathcal{C}_{\text{gold}})_{\text{CWS},\epsilon} = m(\text{TP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}), \text{FP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}), \text{FN}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}})) \quad (22)$$

The sets of certain decisions of $\mathfrak{C}_{\text{Ex}}$ with $\epsilon = 0.2$ are (for illustration see Figure 8):

$$M_{\text{cert.-}0.2}(\mathfrak{C}_{\text{Ex}}) = \{(t_1, t_4), (t_7, t_8)\}, \text{ and}$$

$$U_{\text{cert.-}0.2}(\mathfrak{C}_{\text{Ex}}) = \{(a, b) \mid a, b \in \mathcal{R}_{\text{Ex}}\} - \{(t_1, t_2), (t_1, t_4), (t_2, t_4), (t_5, t_6), (t_7, t_8)\}$$

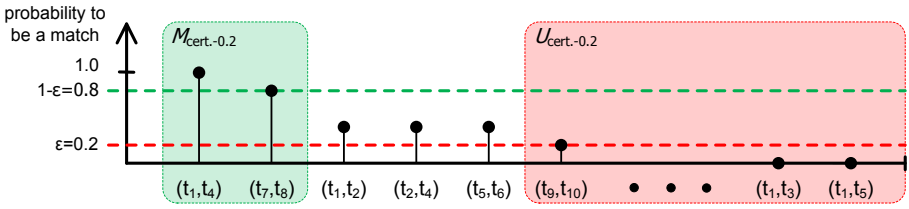


Fig. 8. $M_{\text{cert.-}0.2}(\mathfrak{C}_{\text{Ex}})$ and $U_{\text{cert.-}0.2}(\mathfrak{C}_{\text{Ex}})$ for the tolerance setting $\epsilon = 0.2$

Hence: $\text{TP}_{\text{cert.-}0.2} = \{(t_1, t_4), (t_7, t_8)\}$, $\text{FP}_{\text{cert.-}0.2} = \emptyset$, and $\text{FN}_{\text{cert.-}0.2} = \{(t_3, t_5)\}$. The F_1-score of $\mathfrak{C}_{\text{Ex}}$ using the CWS with the tolerance setting $\epsilon = 0.2$ is thus 0.8.

An important fact of the CWS is that the equations $M = \text{FN} + \text{TP}$ and $U = \text{FP} + \text{TN}$ are not valid anymore and other quality measures, e.g. the number of false decisions (FN + FP), can capture quality aspects which are not captured by precision, recall or F_1-score anymore.

Tolerance Setting. It is to note that a tolerance setting $\epsilon \geq 0.5$ has to be used carefully, because it can lead to inconsistent duplicate clusterings, i.e. a tuple can belong to multiple clusters (cluster-based interpretation) or two tuples can be declared as a match and as an unmatch at the same time (decision-based interpretation).

4 Efficient Quality Computation for the Certain World Semantics

The certain world semantics proposed in the previous section is defined on a complete probabilistic clustering $\mathfrak{C}$. However, as discussed in Section 3.1, instead of $\mathfrak{C}$ usually its factors are available. To rebuild $\mathfrak{C}$ from its factors is most often not practical. For that reason, in this section, we figure out how the quality of a probabilistic clustering $\mathfrak{C}$ can be scored based on the quality scores of its factors.

Cluster-Based Interpretation

Theorem 2. *The certain clustering component of a probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ can be computed by the union of the certain clustering components of its factors:*

$$C_{cert.\text{-}\epsilon}(\mathfrak{C}) = \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} C_{cert.\text{-}\epsilon}(\mathfrak{C}_F) \tag{23}$$

Proof. All factor sets are disjoint. Hence any possible cluster belongs to a single factor. By Definition 5, for every subset (cluster) C of the factor set F of a factor $\mathfrak{C}_F = (\Gamma_F, P_F)$ holds: $\sum_{\mathcal{C} \in \Gamma_F, C \in \mathcal{C}} P_F(\mathcal{C}) = \sum_{\mathcal{C} \in \Gamma, C \in \mathcal{C}} P(\mathcal{C})$. Hence a cluster is certain for $\mathfrak{C}$, if it is certain for its corresponding factor.

Due to Theorem 2, the costs for computing the certain clustering component of a probabilistic clustering can be reduced to the costs required for computing the certain clustering component of its factor with the greatest number of possible clusterings.

Decision-Based Interpretation

Theorem 3. *The set of certain matches $M_{cert.\text{-}\epsilon}$ of a probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ can be computed by the union of the corresponding sets of its factors:*

$$M_{cert.\text{-}\epsilon}(\mathfrak{C}) = \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} M_{cert.\text{-}\epsilon}(\mathfrak{C}_F) \tag{24}$$

Proof. Two tuples can be only a match, i.e. be in a same cluster, if they belong to the same factor set. By Definition 5, for every two tuples $\{t_i, t_j\}$ of the factor set F of a factor $\mathfrak{C}_F = (\Gamma_F, P_F)$ holds: $\sum_{\mathcal{C} \in \Gamma_F, \{t_i,t_j\} \in \mathcal{C}} P_F(\mathcal{C}) = \sum_{\mathcal{C} \in \Gamma, \{t_i,t_j\} \in \mathcal{C}} P(\mathcal{C})$. Hence a tuple pair is certainly a match in $\mathfrak{C}$, if it is certainly a match in its corresponding factor.

Theorem 4. *The decision classes $TP_{cert.\text{-}\epsilon}$ and $FP_{cert.\text{-}\epsilon}$ of a probabilistic clustering $\mathfrak{C} = (\Gamma, P)$ can be computed by the union of the corresponding classes of its factors:*

$$TP_{cert.\text{-}\epsilon}(\mathfrak{C}, \mathcal{C}_{gold}) = \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} TP_{cert.\text{-}\epsilon}(\mathfrak{C}_F, \mathcal{C}_{gold}) \tag{25}$$

$$FP_{cert.\text{-}\epsilon}(\mathfrak{C}, \mathcal{C}_{gold}) = \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} FP_{cert.\text{-}\epsilon}(\mathfrak{C}_F, \mathcal{C}_{gold}) \tag{26}$$

Proof. We prove the theorem only for $\text{TP}_{\text{cert.-}\epsilon}$ ($\text{FP}_{\text{cert.-}\epsilon}$ can be proved accordingly).

$$\begin{aligned}\text{TP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) &= M_{\text{gold}} \cap M_{\text{cert.-}\epsilon}(\mathfrak{C}) \overset{\text{(Theorem 3)}}{=} M_{\text{gold}} \cap \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} M_{\text{cert.-}\epsilon}(\mathfrak{C}_F) \\ &= \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} (M_{\text{gold}} \cap M_{\text{cert.-}\epsilon}(\mathfrak{C}_F)) = \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} \text{TP}_{\text{cert.-}\epsilon}(\mathfrak{C}_F, \mathcal{C}_{\text{gold}})\end{aligned}$$

In contrast to $\text{TP}_{\text{cert.-}\epsilon}$ and $\text{FP}_{\text{cert.-}\epsilon}$, the decision class $\text{FN}_{\text{cert.-}\epsilon}$ cannot be restricted to the individual factors, because it could happen that some true duplicates do not belong to the same factor set. However, we can distinct between inter-factor false negatives (FN^{inter}), i.e. a not detected duplicate pair which tuples belong to different factors (e.g. the tuple pair (t_3, t_5) in our running example) and intra-factor false negatives (FN^{intra}), i.e. a not detected duplicate pair which tuples belong to the same factor.

All inter-factor false negatives are dead certain decisions, because they do not belong to a same cluster in any possible clustering $\mathcal{C} \in \Gamma$. Thus, the set (and hence number) of inter-factor false negatives is the same for all possible clusterings of $\mathfrak{C}$ and can be simply computed from the factor sets: $\text{FN}^{\text{inter}}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) = \text{FN}(\{F \mid \mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})\}, \mathcal{C}_{\text{gold}})$. In contrast, the set of intra-factor false negatives results per definition from the union of the false negative decisions of each factor where each factor $\mathfrak{C}_F$ is only compared with the tuple-equivalent part of the gold standard: $\mathcal{C}^F_{\text{gold}} = \{C \cap F \mid C \in \mathcal{C}_{\text{gold}}\}$.

Using FN^{inter} and $\text{FN}^{\text{intra}}_{\text{cert.-}\epsilon}$, the class of certain false negatives can be computed by:

$$\text{FN}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) = \text{FN}^{\text{inter}}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) \cup \text{FN}^{\text{intra}}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}}) \tag{27}$$

$$= \text{FN}(\{F \mid \mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})\}, \mathcal{C}_{\text{gold}}) \cup \bigcup_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} \text{FN}_{\text{cert.-}\epsilon}(F, \mathcal{C}^F_{\text{gold}}) \tag{28}$$

Thus, the number of certain TP, certain FP and certain FN can be simply computed as:

$$|\mathrm{TP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}})| = \sum_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} |\mathrm{TP}_{\text{cert.-}\epsilon}(\mathfrak{C}_F, \mathcal{C}_{\text{gold}})| \tag{29}$$

$$|\mathrm{FP}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}})| = \sum_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} |\mathrm{FP}_{\text{cert.-}\epsilon}(\mathfrak{C}_F, \mathcal{C}_{\text{gold}})| \tag{30}$$

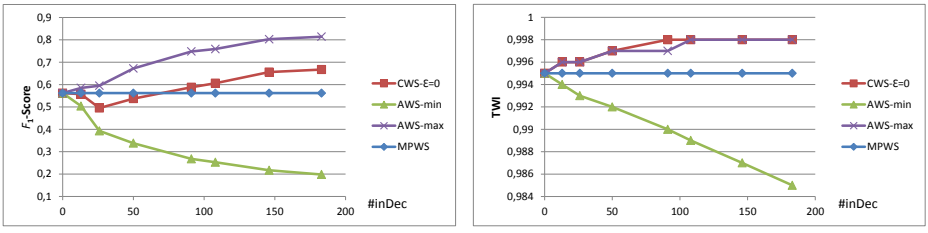
$$|\mathrm{FN}_{\text{cert.-}\epsilon}(\mathfrak{C}, \mathcal{C}_{\text{gold}})| = |\mathrm{FN}^{\text{inter}}(\mathfrak{C}, \mathcal{C}_{\text{gold}})| + \sum_{\mathfrak{C}_F \in \mathcal{F}(\mathfrak{C})} |\mathrm{FN}_{\text{cert.-}\epsilon}(\mathfrak{C}_F, \mathcal{C}^F_{\text{gold}})| \tag{31}$$

In summary, the costs for computing the number of certain TP, certain FP and certain FN and hence the costs for computing the recall, the precision and the F_1-score of the certain facts of a probabilistic clustering can be reduced to the costs required for computing the certain TP, certain FP and certain FN of its factor with the greatest number of possible clusterings.

5 Experimental Evaluation

To experimental evaluate our quality semantics, we use a duplicate detection scenario extensively discussed in [9]. In this scenario, on a real-life CD data set several indeterministic duplicate detection processes have been performed. Each process is characterized by its number of indeterministically handled decisions (#inDec). In Figure 9, we present the F_1-score and the TWI of these processes each scored with our different quality semantics (PWS is not considered, because it does not supply a single value).

The most probable world was the same for all processes and hence the MPWS returned a constant result. Moreover, the quality of the MPWS was equivalent to the quality of a deterministic approach (#inDec =0). The minimal (maximal) possible score was always lower (greater) than by using MPWS and decreased (increased) with growing uncertainty. The CWS without tolerance ($\epsilon = 0$) performed the better than the MPWS, the more uncertainty was modeled in the indeterministic result. Only for less uncertainty, the CWS was worse than MPWS using the F_1-score. In general, the TWI was not that restrictive than the F_1-score (all scores were between 0.984 and 0.998), but show similar results than the F_1-score. Solely the CWS was always better than the MPWS (sometimes even better than the maximal possible quality score).



(a) Decision-based Evaluation (F_1-score) (b) Cluster-based Evaluation (TWI)

Fig. 9. F_1-score and TWI of several indeterministic duplicate detection results

6 Related Work

Quality evaluation of deterministic duplicate detection results has been considered in several works [5,7,11], but none of them take an uncertain clustering into account. De Keijzer et al. propose measures for scoring the quality of uncertain data [3]. The expected precision and the expected recall for scoring the quality of uncertain query results are similar to a variant of our aggregated world semantics. Further semantics, especially the certain world semantics are not covered by this work. Moreover, they restrict themselves to probabilistic results with independent events (in our case, decisions), which is not useful for duplicate detection scenarios.

7 Conclusion

Duplicate detection usually comes along with a high degree of uncertainty and often it cannot be determined with absolute certainty whether two tuples are duplicates or not. Indeterministic duplicate detection approaches have been proposed to handle uncertainty on duplicate decisions by storing multiple possible duplicate clusterings in the resultant data. In this paper, we introduced a framework for scoring the quality of indeterministic duplicate detection results. For that purpose, we presented four different quality semantics, each defined for a special class of data processing tasks.

In this paper, we only went into computation details for a single semantics. In future work, we aim to focus on an efficient computation of the remaining three semantics.

References

1. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent Query Answers in Inconsistent Databases. In: PODS, pp. 68–79 (1999)
2. Beskales, G., Soliman, M.A., Ilyas, I.F., Ben-David, S.: Modeling and Querying Possible Repairs in Duplicate Detection. PVLDB 2(1), 598–609 (2009)
3. de Keijzer, A., van Keulen, M.: Quality Measures in Uncertain Data Management. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 104–115. Springer, Heidelberg (2007)
4. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Trans. Knowl. Data Eng. 19(1), 1–16 (2007)
5. Hassanzadeh, O., Chiang, F., Miller, R.J., Lee, H.C.: Framework for Evaluating Clustering Algorithms in Duplicate Detection. PVLDB 2(1), 1282–1293 (2009)
6. Ioannou, E., Nejdl, W., Niederée, C., Velegrakis, Y.: On-the-Fly Entity-Aware Query Processing in the Presence of Linkage. PVLDB 3(1), 429–438 (2010)
7. Menestrina, D., Whang, S., Garcia-Molina, H.: Evaluating entity resolution results. PVLDB 3(1), 208–219 (2010)
8. Naumann, F., Herschel, M.: An Introduction to Duplicate Detection. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2010)
9. Panse, F., van Keulen, M., Ritter, N.: Indeterministic Handling of Uncertain Decisions in Deduplication. Journal of Data and Information Quality (accepted for publication, 2012)
10. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2011)
11. Talburt, J.R.: Entity Resolution and Information Quality. Morgan Kaufmann (2011)