Data Cleaning und Record Matching

Seminar Informationsintegration und Informationsqualität

Christoph R. Hartel

TU Kaiserslautern

14. Juli 2006

Gliederung

Gliederung

Was können wir bisher?

- Wir können . . .
  - heterogene Datenquellen auffinden,
  - bewerten,
  - ihre Schemata matchen,
  - in ein globales Schema abbilden
  - und uniform darauf zugreifen.

Was können wir bisher?

- Wir können . . .
  - heterogene Datenquellen auffinden,
  - bewerten,
  - ihre Schemata matchen,
  - in ein globales Schema abbilden
  - und uniform darauf zugreifen.

- **Und was fehlt?**

Was können wir bisher?

- Wir können . . .
  - heterogene Datenquellen auffinden,
  - bewerten,
  - ihre Schemata matchen,
  - in ein globales Schema abbilden
  - und uniform darauf zugreifen.

- **Und was fehlt?**
  - **Behandlung der eigentlichen Daten!**
    →Data Cleaning
  - Warum?
    →„garbage in, garbage out“

Was ist Data Cleaning?

Definition (Data Cleaning)

Data Cleaning ist der Prozess der **Identifikation** und **Korrektur** von Anomalien in einer gegebenen Datenmenge.

Definition (Anomalie)

Eine **Anomalie** ist eine Eigenschaft einer Menge von Datensätzen, die dazu führt, dass diese Datensätze eine **falsche Repräsentation der Miniwelt** darstellen.

Was ist Data Cleaning?

Definition (Data Cleaning)

Data Cleaning ist der Prozess der **Identifikation** und **Korrektur** von Anomalien in einer gegebenen Datenmenge.

Definition (Anomalie)

Eine **Anomalie** ist eine Eigenschaft einer Menge von Datensätzen, die dazu führt, dass diese Datensätze eine **falsche Repräsentation der Miniwelt** darstellen.

Gliederung

Datenanomalien

3 Arten von Anomalien:

- **Syntaktische Anomalien**
- **Semantische Anomalien**
- **(Abdeckungsanomalien)**
  - betreffen Vollständigkeit der Daten
    $\rightarrow$Nicht Teil des Data Cleaning im engeren Sinne

Syntaktische Anomalien

- **Lexikalische Fehler**
  - = Fehler in der Struktur der Daten
  - z.B.
    - Misfielded Values
    - Embedded Values
    - ...
- **Formatierungsfehler**
  - = Abweichung von Formatierungskonventionen
  - z.B.
    - Abkürzungen
    - Synonyme
    - Dummy-Werte
    - ...

Semantische Anomalien

- **Verletzung von Integritätsbedingungen**
  - Obermenge der im Schema spezifizierten!
  - z. B. Alter kleiner 0, Verletzung von FA zw. „PLZ“ und „Ort“
- **Fehlerhafte Daten**
  - verletzen keine Integritätsbedingungen, aber decken sich aber nicht mit Eigenschaften des Bezugsobjekts in Miniwelt
  - Typographische Fehler, Konvertierungsfehler, Aliase, . . .
- **Duplikate**
  - z. B. zwei identische Datensätze
  - $\rightarrow$später im Detail!

Gliederung

Probleme des Data Cleanings

- **Größe der Datenmenge**
  - Schema Matching: Anzahl der „Spalten“ ($N \approx 100$)
  - Data Cleaning: Anzahl der „Zeilen“ ($N \approx 100$ Mio.)
- **Verfügbarkeit von Ressourcen**
  - Fast immer enge Schranken für
    - Zeit
    - Hardware
    - Fachpersonal
- **Häufigkeit der Durchführung**
  - Schema Matching erfolgt einmal (bzw. selten)
  - Data Cleaning sollte regelmäßig erfolgen

Probleme des Data Cleanings

- **Größe der Datenmenge**
  - Schema Matching: Anzahl der „Spalten“ ($N \approx 100$)
  - Data Cleaning: Anzahl der „Zeilen“ ($N \approx 100$ Mio.)
- **Verfügbarkeit von Ressourcen**
  - Fast immer enge Schranken für
    - Zeit
    - Hardware
    - Fachpersonal
- **Häufigkeit der Durchführung**
  - Schema Matching erfolgt einmal (bzw. selten)
  - Data Cleaning sollte regelmäßig erfolgen

Probleme des Data Cleanings

- **Größe der Datenmenge**
  - Schema Matching: Anzahl der „Spalten“ ($N \approx 100$)
  - Data Cleaning: Anzahl der „Zeilen“ ($N \approx 100$ Mio.)
- **Verfügbarkeit von Ressourcen**
  - Fast immer enge Schranken für
    - Zeit
    - Hardware
    - Fachpersonal
- **Häufigkeit der Durchführung**
  - Schema Matching erfolgt einmal (bzw. selten)
  - Data Cleaning sollte regelmäßig erfolgen

Probleme des Data Cleanings (2)

- **Vorhandensein von global eindeutigen IDs**
  - Annahme bisher: Equi-Join über globale IDs möglich
  - Kundendatenbanken zweier Unternehmen?
    $\Rightarrow$ Nicht gegeben!
- **Behandlung von Duplikaten**
  - Annahme bisher: Duplikate werden einfach eliminiert
  - Nicht-exakten Duplikaten? Informationsgehalt?
    $\Rightarrow$ Nicht trivial!
- **Manuelle Nachbearbeitung**
  - Annahme bisher: Unklare Datensätze von Hand
  - Für praktische Anwendungen?
    $\Rightarrow$ Illusorisch... (Datenmenge!)

Probleme des Data Cleanings (2)

- **Vorhandensein von global eindeutigen IDs**
  - Annahme bisher: Equi-Join über globale IDs möglich
  - Kundendatenbanken zweier Unternehmen?
    $\Rightarrow$ Nicht gegeben!
- **Behandlung von Duplikaten**
  - Annahme bisher: Duplikate werden einfach eliminiert
  - Nicht-exakten Duplikaten? Informationsgehalt?
    $\Rightarrow$ Nicht trivial!
- **Manuelle Nachbearbeitung**
  - Annahme bisher: Unklare Datensätze von Hand
  - Für praktische Anwendungen?
    $\Rightarrow$ Illusorisch... (Datenmenge!)

Probleme des Data Cleanings (2)

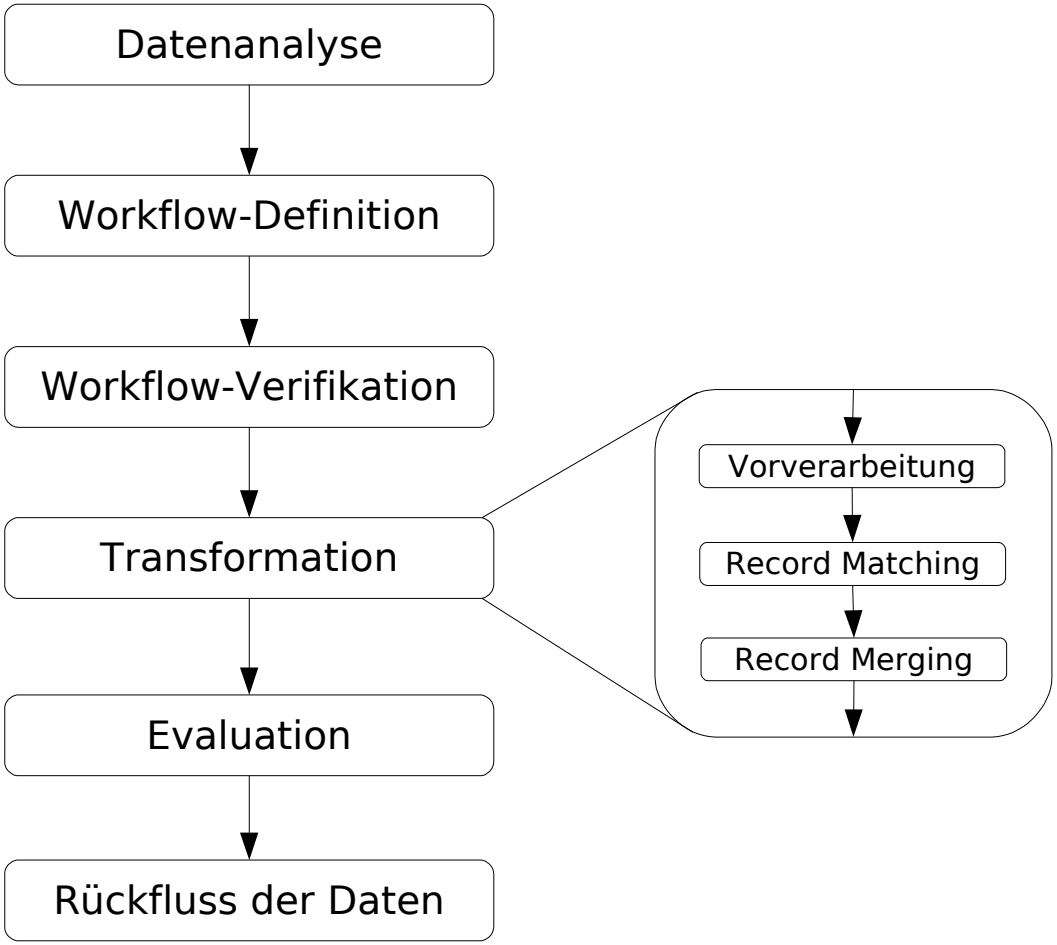
- **Vorhandensein von global eindeutigen IDs**
  - Annahme bisher: Equi-Join über globale IDs möglich
  - Kundendatenbanken zweier Unternehmen?
    $\Rightarrow$ Nicht gegeben!
- **Behandlung von Duplikaten**
  - Annahme bisher: Duplikate werden einfach eliminiert
  - Nicht-exakten Duplikaten? Informationsgehalt?
    $\Rightarrow$ Nicht trivial!
- **Manuelle Nachbearbeitung**
  - Annahme bisher: Unklare Datensätze von Hand
  - Für praktische Anwendungen?
    $\Rightarrow$ Illusorisch... (Datenmenge!)

Gliederung

Ausgangspunkt des Data Cleanings

- Entstehung der Datenbasis ist egal:
  - Integration
  - Existierende Datenbasis
  - ...
- **Annahmen des Data Cleanings:**
  - Daten liegen in einem **einzigen, definierten Schema** vor
  - Die Daten sind im Sinne des Schemas **konsistent**
  - Auf alle Daten ist ein **uniformer Zugriff** möglich **(r/w!)**

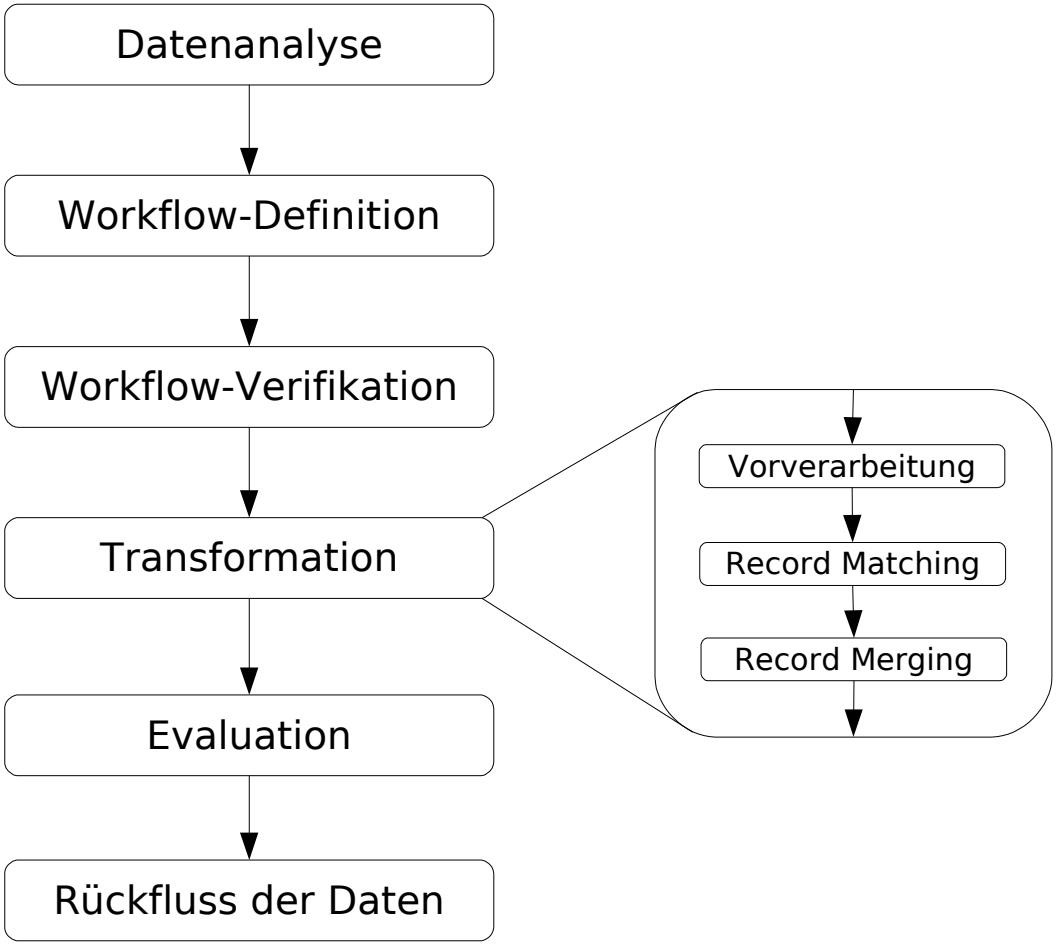
Ablauf des Data Cleanings



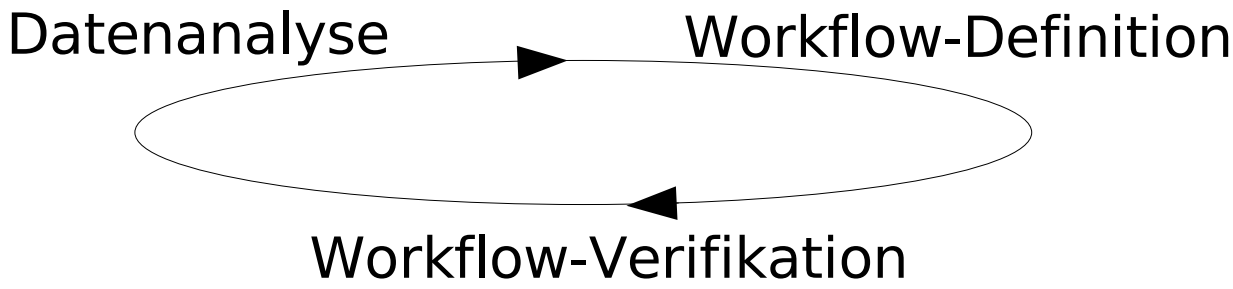
Datenanalyse

- **Ziele:**
  - Gewinnung von (über das Schema hinausgehenden) Metadaten
    Integritätsbedinungen, statistische Merkmale, . . .
  - Identifikation von Anomalien
    Grundlage der Workflow-Spezifikation!
- **Ansätze:**
  - „Einfache Statistik“: Min, Max, Varianz, . . .
  - Pattern-Matching
  - Regeln (für Beziehungen von Attributen)
- **Optimierung:**
  - Wiederverwendung von Analyseergebnissen bei Schema Matching!

Ablauf des Data Cleanings: Definition & Verifikation

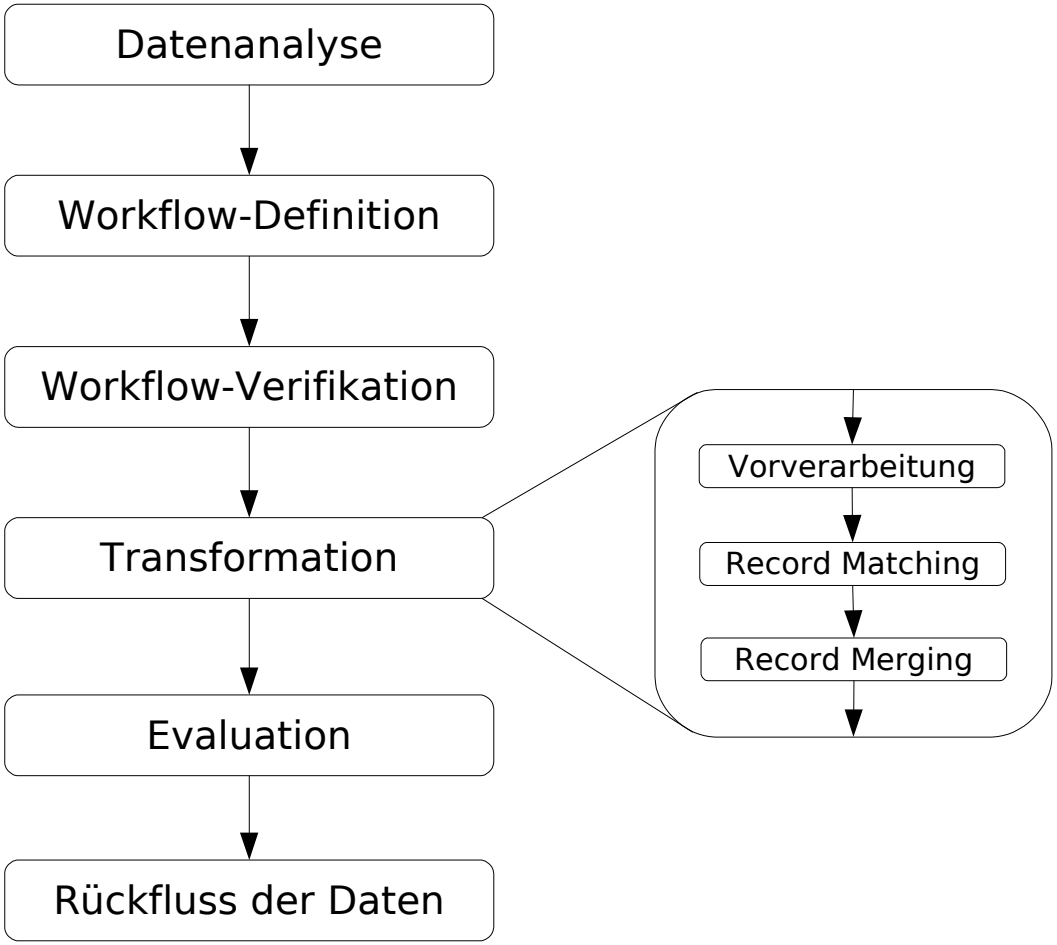


Workflow-Definition und -Verfikation



- **Interessanter Ansatz:** Interaktive Spezifikation
  - Immediate Feedback
  - Highlighting der zu ändernden Werte
  - Undo-Funktionalität
  - Spezifikation durch Beispiel
  - *Potter's Wheel*

Ablauf des Data Cleanings: Vorverarbeitung



Datenvorverarbeitung

⇒ **Beseitigung aller Anomalien bis auf Duplikate**

- **Normalisierung** →syntaktische Anomalien
  - *Attribute Split* (z.B. „Adresse“) →Schemaebene?
  - *Standardisierung* (Abkürzungen, Termreihenfolge, . . .)
  - Ist Normalisierung immer eine gute Idee? (Informationsverlust?, Uneinheitliche Daten?, . . .)
- **Validierung** →semantische Anomalien
  - *Typographische Fehler*, z. B. „Hasn“ statt „Hans“
  - *Ausreißer*, z. B. Geburtsjahr „1897“ statt „1987“
  - *Inkonsistenzen*, z. B. zw. „PLZ“ und „Ort“
  - . . .

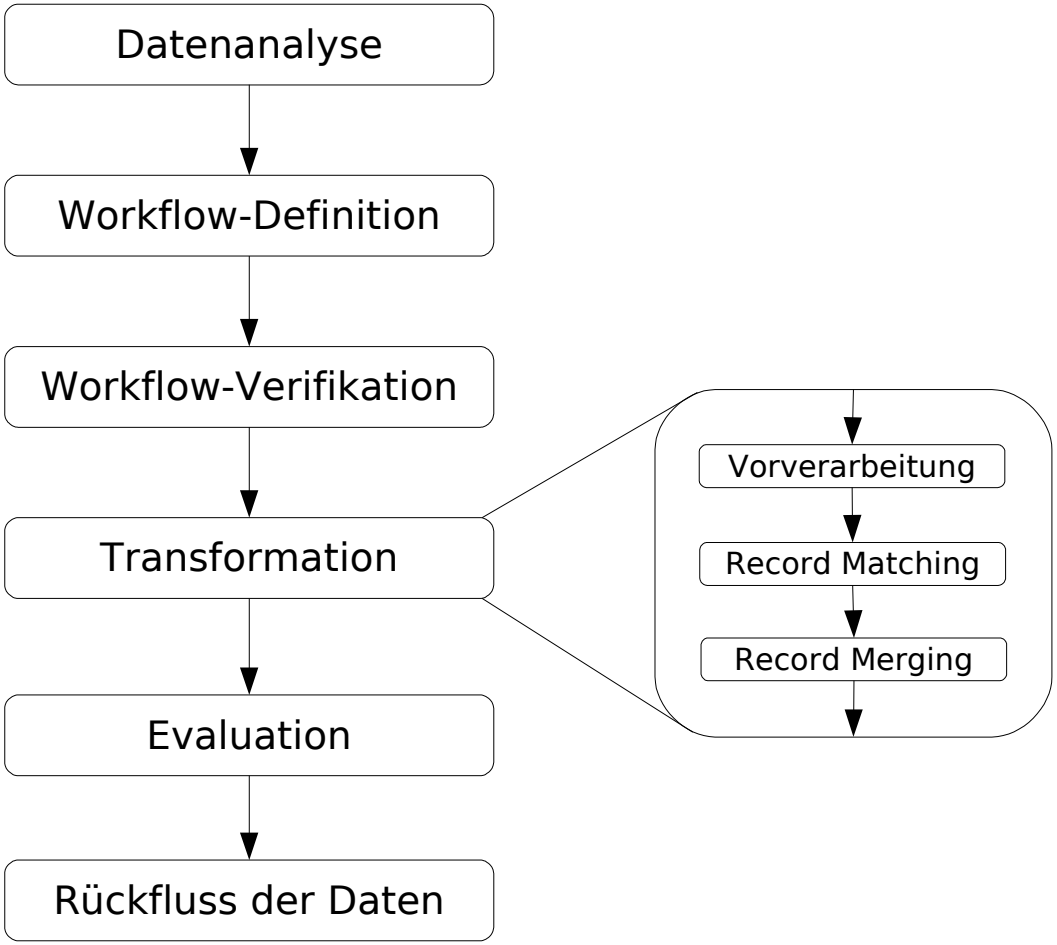
Datenvorverarbeitung

⇒ **Beseitigung aller Anomalien bis auf Duplikate**

- **Normalisierung** →syntaktische Anomalien
  - *Attribute Split* (z.B. „Adresse“) →Schemaebene?
  - *Standardisierung* (Abkürzungen, Termreihenfolge, . . .)
  - Ist Normalisierung immer eine gute Idee? (Informationsverlust?, Uneinheitliche Daten?, . . .)
- **Validierung** →semantische Anomalien
  - *Typographische Fehler*, z. B. „Hasn“ statt „Hans“
  - *Ausreißer*, z. B. Geburtsjahr „1897“ statt „1987“
  - *Inkonsistenzen*, z. B. zw. „PLZ“ und „Ort“
  - . . .

Gliederung

Ablauf des Data Cleanings: Record Matching



Was ist Record Matching?

Definition (Record Matching)

Record Matching ist die **Identifikation** von Duplikaten in einer Menge von Datensätzen.

Alternativ: Record Linkage, Object Identification, Entity Resolution, Reference Reconciliation, ...

Definition (Duplikat/Äquivalenz)

Ein Datensatz R_1 ist ein **Duplikat** eines anderen Datensatzes R_2 (mit $R_1 \neq R_2$), wenn beide Datensätze **dasselbe Bezugsobjekt** in der Miniwelt repräsentieren.

Annahme: Jeder Datensatz hat genau ein Bezugsobjekt.

Was ist Record Matching?

Definition (Record Matching)

Record Matching ist die **Identifikation** von Duplikaten in einer Menge von Datensätzen.

Alternativ: Record Linkage, Object Identification, Entity Resolution, Reference Reconciliation, ...

Definition (Duplikat/Äquivalenz)

Ein Datensatz R_1 ist ein **Duplikat** eines anderen Datensatzes R_2 (mit $R_1 \neq R_2$), wenn beide Datensätze **dasselbe Bezugsobjekt** in der Miniwelt repräsentieren.

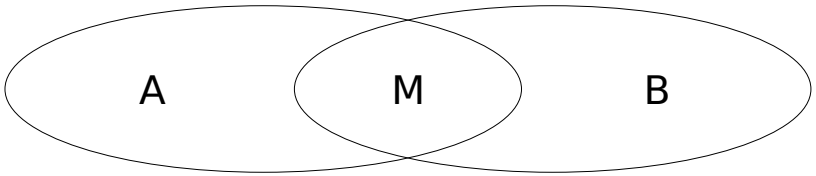
Annahme: Jeder Datensatz hat genau ein Bezugsobjekt.

Modell von Fellegi & Sunter – Überblick

- Fellegi und Sunter definieren 1969 Modell für Record Matching
- Liefert theoretische Fundierung für alle heutigen Verfahren
- Abstraktes Modell: *keine* Aussagen über Realisierung
- **Ausgangspunkt**
  - A, B zwei Mengen von Datensätzen
  - Matches M (selbes Bezugsobjekt), Nicht-Matches U
  - M und U existieren, aber sind unbekannt!

Modell von Fellegi & Sunter – Überblick

- Fellegi und Sunter definieren 1969 Modell für Record Matching
- Liefert theoretische Fundierung für alle heutigen Verfahren
- Abstraktes Modell: *keine* Aussagen über Realisierung
- **Ausgangspunkt**



- A, B zwei Mengen von Datensätzen
- Matches M (selbes Bezugsobjekt), Nicht-Matches U
- M und U existieren, aber sind unbekannt!

Modell von Fellegi & Sunter – Ablauf

- **1. Schritt:** Vergleichsraum Γ definieren
  - Γ wird durch beliebige Kriterien aufgespannt, z.B.:
    - „Vorname stimmt überein“
    - „Nachname ist ähnlich“
    - „Geburtsdatum weicht um max. 2 Jahre ab“
- **2. Schritt:** Vergleichsfunktion $comp : A \times B \longrightarrow \Gamma$
  - z.B. $comp(R_1, R_2) = (1, 1, 0) = \gamma \in \Gamma$
- **3. Schritt:** Entscheidungsfunktion $dec : \Gamma \longrightarrow \{L, NL, PL\}$
  - Links (L) $\neq$ Matches, Nicht-Links (NL), mögliche Links (PL)

$$dec(\gamma) = \begin{cases} L, & \text{falls } r(\gamma) > t_{upper} \\ NL, & \text{falls } r(\gamma) < t_{lower} \\ PL, & \text{sonst} \end{cases}$$

  - $PL \Rightarrow$ Clerical Review; $r =$ Agreement Ratio

Modell von Fellegi & Sunter – Ablauf

- **1. Schritt:** Vergleichsraum Γ definieren
  - Γ wird durch beliebige Kriterien aufgespannt, z.B.:
    - „Vorname stimmt überein“
    - „Nachname ist ähnlich“
    - „Geburtsdatum weicht um max. 2 Jahre ab“
- **2. Schritt:** Vergleichsfunktion $comp : A \times B \longrightarrow \Gamma$
  - z.B. $comp(R_1, R_2) = (1, 1, 0) = \gamma \in \Gamma$
- **3. Schritt:** Entscheidungsfunktion $dec : \Gamma \longrightarrow \{L, NL, PL\}$
  - Links (L) $\neq$ Matches, Nicht-Links (NL), mögliche Links (PL)

$$dec(\gamma) = \begin{cases} L, & \text{falls } r(\gamma) > t_{upper} \\ NL, & \text{falls } r(\gamma) < t_{lower} \\ PL, & \text{sonst} \end{cases}$$

  - $PL \Rightarrow$ Clerical Review; $r =$ Agreement Ratio

Modell von Fellegi & Sunter – Ablauf

- **1. Schritt:** Vergleichsraum Γ definieren
  - Γ wird durch beliebige Kriterien aufgespannt, z.B.:
    - „Vorname stimmt überein“
    - „Nachname ist ähnlich“
    - „Geburtsdatum weicht um max. 2 Jahre ab“
- **2. Schritt:** Vergleichsfunktion $comp : A \times B \longrightarrow \Gamma$
  - z.B. $comp(R_1, R_2) = (1, 1, 0) = \gamma \in \Gamma$
- **3. Schritt:** Entscheidungsfunktion $dec : \Gamma \longrightarrow \{L, NL, PL\}$
  - Links (L) $\neq$ Matches, Nicht-Links (NL), mögliche Links (PL)

$$dec(\gamma) = \begin{cases} L, & \text{falls } r(\gamma) > t_{upper} \\ NL, & \text{falls } r(\gamma) < t_{lower} \\ PL, & \text{sonst} \end{cases}$$

  - $PL \Rightarrow$ Clerical Review; $r =$ Agreement Ratio

Record-Matching-Verfahren

- In der Literatur ex. (scheinbar) zahlreiche Verfahren
- Aber:
  Verfahren = Ablaufsteuerung + Ähnlichkeitsmetriken
- **Ähnlichkeit** $sim(R_1, R_2) = \sum w_i \cdot sim(v_i^{R_1}, v_i^{R_2}) \in [0, \ldots, 1]$
  - Metriken
    - *Zahlen* (z.B. Geburtsjahr)
      $\rightarrow$Vergleich sehr einfach, aber Aussagekraft?
    - *Strings* (z.B. Name)
      $\rightarrow$Editierabstände, Phonetik, Abkürzungen, WHIRL, ...
    - *Konstanten*?
      z.B. Geschlecht „Männlich“ / „Weiblich“ vs. „M“ / „F“ vs. 0 / 1
      $\rightarrow$Vorarbeit in Analysephase notwendig!

Record-Matching-Verfahren

- In der Literatur ex. (scheinbar) zahlreiche Verfahren
- Aber:
  Verfahren $=$ Ablaufsteuerung $+$ Ähnlichkeitsmetriken

- **Ähnlichkeit** $sim(R_1, R_2) = \sum w_i \cdot sim(v_i^{R_1}, v_i^{R_2}) \in [0, \ldots, 1]$
  - Metriken
    - *Zahlen* (z.B. Geburtsjahr)
      $\rightarrow$Vergleich sehr einfach, aber Aussagekraft?
    - *Strings* (z.B. Name)
      $\rightarrow$Editierabstände, Phonetik, Abkürzungen, WHIRL, ...
    - *Konstanten*?
      z.B. Geschlecht „Männlich“ / „Weiblich“ vs. „M“ / „F“ vs. 0 / 1
      $\rightarrow$Vorarbeit in Analysephase notwendig!

Record-Matching-Verfahren

- In der Literatur ex. (scheinbar) zahlreiche Verfahren
- Aber:
  Verfahren $=$ Ablaufsteuerung $+$ Ähnlichkeitsmetriken

- **Ähnlichkeit** $sim(R_1, R_2) = \sum w_i \cdot sim(v_i^{R_1}, v_i^{R_2}) \in [0, \ldots, 1]$
  - Metriken
    - *Zahlen* (z.B. Geburtsjahr)
      $\rightarrow$Vergleich sehr einfach, aber Aussagekraft?
    - *Strings* (z.B. Name)
      $\rightarrow$Editierabstände, Phonetik, Abkürzungen, WHIRL, ...
    - *Konstanten*?
      z.B. Geschlecht „Männlich“ / „Weiblich“ vs. „M“ / „F“ vs. 0 / 1
      $\rightarrow$Vorarbeit in Analysephase notwendig!

Record-Matching-Verfahren

- In der Literatur ex. (scheinbar) zahlreiche Verfahren
- Aber:
  Verfahren $=$ Ablaufsteuerung $+$ Ähnlichkeitsmetriken

- **Ähnlichkeit** $sim(R_1, R_2) = \sum w_i \cdot sim(v_i^{R_1}, v_i^{R_2}) \in [0, \ldots, 1]$

  - Metriken
    - *Zahlen* (z.B. Geburtsjahr)
      $\rightarrow$Vergleich sehr einfach, aber Aussagekraft?
    - *Strings* (z.B. Name)
      $\rightarrow$Editierabstände, Phonetik, Abkürzungen, WHIRL, ...
    - *Konstanten*?
      z.B. Geschlecht „Männlich“ / „Weiblich“ vs. „M“ / „F“ vs. 0 / 1
      $\rightarrow$Vorarbeit in Analysephase notwendig!

Record Matching: Ablaufsteuerung

- Algorithmen angelehnt an JOIN-Implementierungen
- **Naiver Algorithmus: Nested Loops**
  - Vergleiche jeden Datensatz mit jedem anderen $\Rightarrow O(N^2)$
- **Sorted-Neighbourhood-Verfahren**
  - Erweitert Idee von Sort-Merge-JOIN
  - *Probleme:*
    - Keine eindeutigen IDs
    - Daten potentiell fehlerhaft

Sorted-Neighbourhood-Verfahren

1. Schlüssel berechnen

Name	Vorname	GebDat
Maier	Hans	27.03.1974
Schmitt	Elisabeth	04.11.1980
Maier	Hans	27.03.1947
Müller	Karl	15.06.1958
Schmitt	Maria	18.09.1963
Maier	Hasn	27.03.1974
Mayer	Franz	21.12.1971
Müller	Günther	03.02.1984
Schmidt	Ernst	29.08.1967

Sorted-Neighbourhood-Verfahren

2. Nach Schlüssel sortieren

Name	Vorname	GebDat	Schlüssel
Maier	Hans	27.03.1947	47Maie
Müller	Karl	15.06.1958	58Müll
Schmitt	Maria	18.09.1963	63Schm
Schmidt	Ernst	29.08.1967	67Schm
Mayer	Franz	21.12.1971	71Maye
Maier	Hans	27.03.1974	74Maie
Maier	Hasn	27.03.1974	74Maie
Schmitt	Elisabeth	04.11.1980	80Schm
Müller	Günther	03.02.1984	84Müll

Sorted-Neighbourhood-Verfahren

3. „Fenster“ über Daten schieben (Iterativ)

- Fixe Fenstergröße, z.B. $w = 3$

Name	Vorname	GebDat	Schlüssel
Maier	Hans	27.03.1947	47Maie
Müller	Karl	15.06.1958	58Müll
Schmitt	Maria	18.09.1963	63Schm
Schmidt	Ernst	29.08.1967	67Schm
Mayer	Franz	21.12.1971	71Maye
Maier	Hans	27.03.1974	74Maie
Maier	Hasn	27.03.1974	74Maie
Schmitt	Elisabeth	04.11.1980	80Schm
Müller	Günther	03.02.1984	84Müll

Sorted-Neighbourhood-Verfahren

4. Pro Iteration: Alle R im Fenster vergleichen

- Nested Loop, aber für kleines N
  $\Rightarrow O(NlogN)$ für $w \ll N$ (genauer: $w < \lceil logN \rceil$)

Name	Vorname	GebDat	Schlüssel
Maier	Hans	27.03.1947	47Maie
Müller	Karl	15.06.1958	58Müll
Schmitt	Maria	18.09.1963	63Schm
Schmidt	Ernst	29.08.1967	67Schm
Mayer	Franz	21.12.1971	71Maye
Maier	Hans	27.03.1974	74Maie
Maier	Hasn	27.03.1974	74Maie
Schmitt	Elisabeth	04.11.1980	80Schm
Müller	Günther	03.02.1984	84Müll

Sorted-Neighbourhood-Verfahren

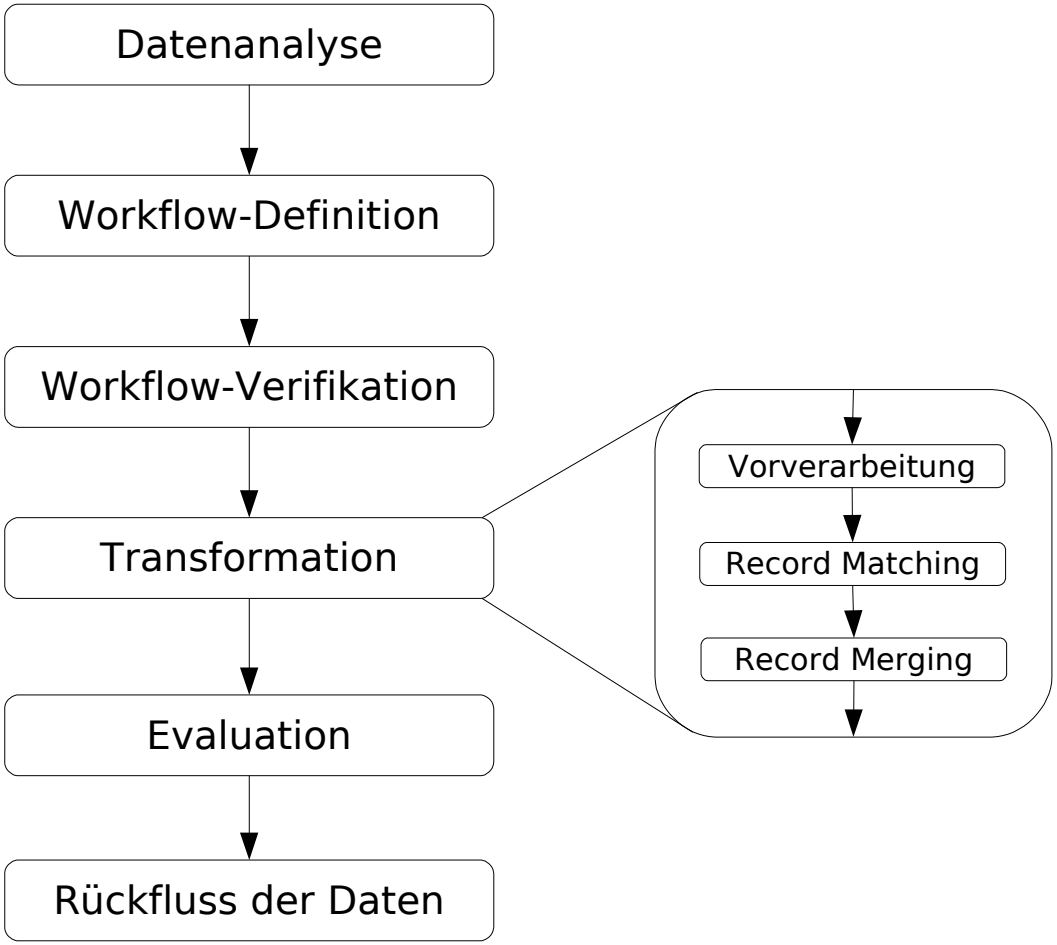
Problem: Fehlertoleranz sehr gering!

- Bei Fehler in Schlüssel-Attributen Einordnung in falsche Nachbarschaft

Name	Vorname	GebDat	Schlüssel
Maier	Hans	27.03.1947	47Maie
Müller	Karl	15.06.1958	58Müll
Schmitt	Maria	18.09.1963	63Schm
Schmidt	Ernst	29.08.1967	67Schm
Mayer	Franz	21.12.1971	71Maye
Maier	Hans	27.03.1974	74Maie
Maier	Hasn	27.03.1974	74Maie
Schmitt	Elisabeth	04.11.1980	80Schm
Müller	Günther	03.02.1984	84Müll

Gliederung

Ablauf des Data Cleanings: Record Merging



Grundlagen des Record Merging

Definition (Record Merging)

Record Merging ist die **Behandlung** von Duplikaten in einer Menge von Datensätzen.

- Trivialer Fall: Eliminierung von Duplikaten
  - Aber: Ist das immer eine gute Idee?
- **Erster Schritt:**
  - In welcher Beziehung können äquivalente Datensätze zueinander stehen?
- **Zweiter Schritt:**
  - Wie können wir damit jeweils umgehen?

Grundlagen des Record Merging

Definition (Record Merging)

Record Merging ist die **Behandlung** von Duplikaten in einer Menge von Datensätzen.

- Trivialer Fall: Eliminierung von Duplikaten
  - Aber: Ist das immer eine gute Idee?
- **Erster Schritt:**
  - In welcher Beziehung können äquivalente Datensätze zueinander stehen?
- **Zweiter Schritt:**
  - Wie können wir damit jeweils umgehen?

Grundlagen des Record Merging

Definition (Record Merging)

Record Merging ist die **Behandlung** von Duplikaten in einer Menge von Datensätzen.

- Trivialer Fall: Eliminierung von Duplikaten
  - Aber: Ist das immer eine gute Idee?
- **Erster Schritt:**
  - In welcher Beziehung können äquivalente Datensätze zueinander stehen?
- **Zweiter Schritt:**
  - Wie können wir damit jeweils umgehen?

1. Beziehungen äquivalenter Datensätze

R_1: Name = Maier, Vorname = Hans, Gehalt = 42.000
R_2: Name = Maier, Vorname = Hans, Gehalt = 42.000
R_3: Name = Maier, Vorname = Hans, Gehalt = 42.000, Durchwahl = 1234
R_4: Name = Maier, Vorname = Hans, Gehalt = 24.000

- 3 Fälle:
  1. **Identität:** Gleiche Attributmenge (Spalten) und gleiche Werte („Exaktes Duplikat“)
  2. **Komplementarität:** Mindestens ein *nicht* gemeinsames Attribut
  3. **Konflikt:** Mindestens ein gemeinsames Attribut, dessen Wert sich unterscheidet
- Fall 1 schließt die anderen beiden aus
- Fall 2 und Fall 3 kompatibel

1. Beziehungen äquivalenter Datensätze

R_1: Name = Maier, Vorname = Hans, Gehalt = 42.000
R_2: Name = Maier, Vorname = Hans, Gehalt = 42.000
R_3: Name = Maier, Vorname = Hans, Gehalt = 42.000, Durchwahl = 1234
R_4: Name = Maier, Vorname = Hans, Gehalt = 24.000

- 3 Fälle:
  1. **Identität:** Gleiche Attributmenge (Spalten) und gleiche Werte („Exaktes Duplikat“)
  2. **Komplementarität:** Mindestens ein *nicht* gemeinsames Attribut
  3. **Konflikt:** Mindestens ein gemeinsames Attribut, dessen Wert sich unterscheidet
- Fall 1 schließt die anderen beiden aus
- Fall 2 und Fall 3 kompatibel

2. Behandlung äquivalenter Datensätze

- **Identität**
  $\rightarrow$Trivial (Duplikat-Eliminierung)
- **Komplementarität**
  $\rightarrow$Übertragen der zusätzlichen Daten, dann wie Identität (Sonderfall: NULL-Werte)
- **Konflikt**
  $\rightarrow$Nicht trivial!
  - *Ignoranz* (Datensätze bleiben unverändert)
  - *Vermeidung* (Mengenwertige Attribute, Masking)
    $\Rightarrow$ Verlagerung des Problems in die Anfragezeit . . .
  - *Auflösung!*

2. Behandlung äquivalenter Datensätze

- **Identität**
  $\longrightarrow$Trivial (Duplikat-Eliminierung)
- **Komplementarität**
  $\longrightarrow$Übertragen der zusätzlichen Daten, dann wie Identität (Sonderfall: NULL-Werte)
- **Konflikt**
  $\longrightarrow$Nicht trivial!
  - *Ignoranz* (Datensätze bleiben unverändert)
  - *Vermeidung* (Mengenwertige Attribute, Masking)
    $\Rightarrow$ Verlagerung des Problems in die Anfragezeit . . .
  - *Auflösung!*

2. Behandlung äquivalenter Datensätze

- **Identität**
  $\longrightarrow$Trivial (Duplikat-Eliminierung)
- **Komplementarität**
  $\longrightarrow$Übertragen der zusätzlichen Daten, dann wie Identität (Sonderfall: NULL-Werte)
- **Konflikt**
  $\longrightarrow$Nicht trivial!
  - *Ignoranz* (Datensätze bleiben unverändert)
  - *Vermeidung* (Mengenwertige Attribute, Masking)
    $\Rightarrow$ Verlagerung des Problems in die Anfragezeit . . .
  - *Auflösung!*

Auflösung von Konflikten

- Ersetzung konfliktärer Attributwerte durch je einen einzigen, semantisch sinnvollen Wert

- **Selektion / Aggregation**
  $\rightarrow$Wähle einen Werte aus bzw. berechne neuen Wert
  - Voting, wahrscheinlichster Wert, ...
  - Durchschnitt, anwendungsspezifische Funktion, ...
  - Sinnvolles Ergebnis?

- **Konfidenz-basiert**
  - Konfidenz = Vertrauen in die Korrektheit von Datensätzen
  - z.B. $R_1.c = 80\%, R_2.c = 70\% \Rightarrow$ Wähle Wert von R_1
  - Komplexe Verwaltungslogik erforderlich!

Auflösung von Konflikten

- Ersetzung konfliktärer Attributwerte durch je einen einzigen, semantisch sinnvollen Wert

- **Selektion / Aggregation**
  $\rightarrow$Wähle einen Werte aus bzw. berechne neuen Wert
  - Voting, wahrscheinlichster Wert, ...
  - Durchschnitt, anwendungsspezifische Funktion, ...
  - Sinnvolles Ergebnis?

- **Konfidenz-basiert**
  - Konfidenz = Vertrauen in die Korrektheit von Datensätzen
  - z.B. $R_1.c = 80\%, R_2.c = 70\% \Rightarrow$ Wähle Wert von R_1
  - Komplexe Verwaltungslogik erforderlich!

Gliederung

Qualitätskriterien

- Qualitätsbetrachtungen oft beschränkt auf Ergebnis
- Für Praxistauglichkeit aber auch der *Prozess* sehr wichtig!

- Daher 2 Arten von Kriterien:
  - **Prozessbezogene Kriterien**
  - **Ergebnisbezogene Kriterien**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Prozessbezogene Qualitätskriterien

- **Algorithmische Komplexität** →min
- **Laufzeit** →min
  - Parallelisierbarkeit, Inkrementelles Cleaning
- **Erforderliche Benutzerinteraktion** →min
  - Spezifikation, Clerical Review
- **Data Lineage** →max
  - Nachvollziehbarkeit der Datenentstehung, ggf. Undo
- **Bezug zur Anwendungsdomäne** →min/max?
- **Wahrung der Privatsphäre** →min/max?
  - Gefahr der Deanonymisierung
  - Allgemeines Problem der Datenintegration!

⇒ **Messbarkeit schwierig!**

Ergebnisbezogene Qualitätskriterien

- **Qualitätsmaße ex. nur für Matching-Ergebnisse**
  - Erfasst indirekt auch Analyse und Vorverarbeitung
  - Aber Qualität des Mergings?!

- **Idee: Vergleich von Links und Matches**
  - Erinnerung:
    - Matches M = Tatsächlich vorhanden
    - Links L = Ergebnis der Klassifizierung
      $\Rightarrow$ Optimal: Links = Matches (d.h. 100% korrekt klassifiziert)
  - **True Positive:** Link, der auch Match ist (OK)
  - **True Negative:** Nicht-Link, der auch Nicht-Match ist (OK)
  - **False Positive:** Link, der nicht Match ist (NOK)
  - **False Negative:** Nicht-Link, der Match ist (NOK)

Ergebnisbezogene Qualitätskriterien

- **Qualitätsmaße ex. nur für Matching-Ergebnisse**
  - Erfasst indirekt auch Analyse und Vorverarbeitung
  - Aber Qualität des Mergings?!
- **Idee: Vergleich von Links und Matches**
  - Erinnerung:
    - Matches M = Tatsächlich vorhanden
    - Links L = Ergebnis der Klassifizierung
      $\Rightarrow$ Optimal: Links = Matches (d.h. 100% korrekt klassifiziert)
  - **True Positive:** Link, der auch Match ist (OK)
  - **True Negative:** Nicht-Link, der auch Nicht-Match ist (OK)
  - **False Positive:** Link, der nicht Match ist (NOK)
  - **False Negative:** Nicht-Link, der Match ist (NOK)

Ergebnisbezogene Qualitätskriterien

- **Qualitätsmaße ex. nur für Matching-Ergebnisse**
  - Erfasst indirekt auch Analyse und Vorverarbeitung
  - Aber Qualität des Mergings?!

- **Idee: Vergleich von Links und Matches**
  - Erinnerung:
    - Matches M = Tatsächlich vorhanden
    - Links L = Ergebnis der Klassifizierung
      $\Rightarrow$ Optimal: Links = Matches (d.h. 100% korrekt klassifiziert)
  - **True Positive:** Link, der auch Match ist (OK)
  - **True Negative:** Nicht-Link, der auch Nicht-Match ist (OK)
  - **False Positive:** Link, der nicht Match ist (NOK)
  - **False Negative:** Nicht-Link, der Match ist (NOK)

Ergebnisbezogene Qualitätskriterien (2)

- **Probleme:**
  - **Bestimmung der Matches**
    - Bei realen Daten (Evaluation) praktisch unmöglich...
      $\Rightarrow$ Generierte Testdaten nötig („Gold Standard Set“)
    - Aber: keine standardisierten Testdaten vorhanden
      $\Rightarrow$ keine vergleichbaren Ergebnisse!
  - **Annahme eines optimalen Schwellwertes t**
    - Bestimmung: eine Ausführung mit *jedem* möglichen t...
      $\Rightarrow$ Praktisch unmöglich
    - Alternativ: Erfahrungswerte, Heuristiken
  - **„Badness“ von FP und FN anwendungsabhängig**
    - Somit keine absoluten Aussagen über Verfahren möglich

Ergebnisbezogene Qualitätskriterien (2)

- **Probleme:**
  - **Bestimmung der Matches**
    - Bei realen Daten (Evaluation) praktisch unmöglich...
      $\Rightarrow$ Generierte Testdaten nötig („Gold Standard Set“)
    - Aber: keine standardisierten Testdaten vorhanden
      $\Rightarrow$ keine vergleichbaren Ergebnisse!
  - **Annahme eines optimalen Schwellwertes** t
    - Bestimmung: eine Ausführung mit *jedem* möglichen t...
      $\Rightarrow$ Praktisch unmöglich
    - Alternativ: Erfahrungswerte, Heuristiken
  - **„Badness“ von FP und FN anwendungsabhängig**
    - Somit keine absoluten Aussagen über Verfahren möglich

Ergebnisbezogene Qualitätskriterien (2)

- **Probleme:**
  - **Bestimmung der Matches**
    - Bei realen Daten (Evaluation) praktisch unmöglich...
      $\Rightarrow$ Generierte Testdaten nötig („Gold Standard Set“)
    - Aber: keine standardisierten Testdaten vorhanden
      $\Rightarrow$ keine vergleichbaren Ergebnisse!
  - **Annahme eines optimalen Schwellwertes** t
    - Bestimmung: eine Ausführung mit *jedem* möglichen t...
      $\Rightarrow$ Praktisch unmöglich
    - Alternativ: Erfahrungswerte, Heuristiken
  - **„Badness“ von FP und FN anwendungsabhängig**
    - Somit keine absoluten Aussagen über Verfahren möglich

Gliederung

Frameworks und Werkzeuge

- **Open Source**
  - AJAX, Febrl, Potter's Wheel, ...
  - Hauptsächlich Forschungsprototypen
    $\Rightarrow$ Interessante Ideen, beschränkte Anwendbarkeit
- **Kommerzielle Lösungen**
  - Großer Markt für Produkte und Dienstleistungen!
  - Zahlreiche **„kleine" Lösungen** für Small Business
    MatchIT, Clean&Match, LinkageWiz, ...
  - **„Große" Lösungen**
    $\rightarrow$Die üblichen Verdächtigen ;)
    IBM, Oracle, (Microsoft?)

Zusammenfassung & Ausblick

- **Zusammenfassung**
  - Data Cleaning = Prozess der Identifikation und Korrektur von Anomalien in einer gegebenen Datenmenge
  - Hauptproblem: Größe der Datenbasis
  - Merging von Datensätzen nicht trivial!
  - Bewertung schwierig, keine einheitlichen Standards

- **Ausblick**
  - Säuberung von nicht-textuellen Daten? (Bilder, Videos, ...)
  - Verfahren, die keine relationalen Schemata voraussetzen? (insbes. für XML)
  - Standards für Bewertung und Vergleich von Verfahren?

Fragen?

Ergänzung

Syntaktische Anomalien

- **Lexikalische Fehler**
  - *fehlerhaft zugeordnete Werte* (engl. misfielded values)
    Werte in „Name“ und „Vorname“ vertauscht
  - *eingebettete Werte*
    „Adresse“ statt {„Straße“, „Hausnummer“, „PLZ“, „Ort“ }
- **Formatierungsfehler**
  - „Vorname Nachname“ statt „Nachname, Vorname“
  - Abkürzungen („Fa.“ statt „Firma“)
  - Synonyme („Entwickler“ vs. „Programmierer“)
  - Dummy-Werte statt eines NULL-Wertes („999“ statt NULL)
  - unterschiedliche Darstellungsformen von Konstanten
    („Männlich“ / „Weiblich“ vs. „M“ / „F“ vs. 0 / 1)
  - . . .

Semantische Anomalien

- **Verletzung von Integritätsbedingungen**
  - Obermenge der im Schema spezifizierten!
  - z. B. Alter kleiner 0, Verletzung von FA zw. „PLZ“ und „Ort“
- **Fehlerhafte Daten**
  - verletzen keine Integritätsbedingungen, aber decken sich aber nicht mit Eigenschaften des Bezugsobjekts in Miniwelt
  - Typographische Fehler („Müllre“ statt „Müller“)
  - Konvertierungsfehler („Müller“ statt „Müller“)
  - Vorsätzliche Verschleierung (etwa Aliase)
  - Unterschiedliche Interpretation von Werten (Euro statt Pfund)
  - . . .
- **Duplikate** →Siehe Record Matching!

Multi-Pass Sorted-Neighbourhood-Verfahren

- **Problem des normalen SNV:**
  - Starke Abhängigkeit von berechnetem Schlüssel
    $\Rightarrow$ geringe Fehlertoleranz
- **Optimierung: MP-SNV**
  - Mehrere Läufe mit unabhängigen Schlüsseln
- **Bewertung**
  - $+$ Kleine Fenstergrößen, daher trotzdem relativ effizient
  - $+$ Verbesserte Genauigkeit bei sehr „unsauberen“ Daten
  - $+$ Ermöglicht Parallelisierung!
  - $-$ Höhere Komplexität als einzelner Lauf

Sonderfall: NULL-Werte

R_2: Name = Maier, Vorname = Hans, Gehalt = 42.000, Durchwahl = NULL

R_3: Name = Maier, Vorname = Hans, Gehalt = 42.000, Durchwahl = 1234

- **Kontext:** Merging komplementärer Daten
- **Problem: Wie ist NULL-Wert entstanden?**
  - durch Überführung in gemeinsames Schema?
  - tatsächlicher (beabsichtigter) NULL-Wert
- Wäre z.B. in XML kein Problem

Ignoranz und Vermeidung von Konflikten

- **Ignoranz**
  - Konfliktäre Datensätze bleiben unverändert erhalten
  - − Verlagerung des Problems in die Anfragezeit . . .

- **Vermeidung**
  - Zusammenführung der Datensätze mit *Mengen-wertigen Attributen* evtl. *Maskierung*
  - − Immer noch Verlagerung des Problems in die Anfragezeit
  - − Komplexe Verwaltungslogik in der Datenquelle erforderlich
  - + Immerhin: Unterstützt Benutzer Umgang mit Konflikten

Ignoranz und Vermeidung von Konflikten

- **Ignoranz**
  - Konfliktäre Datensätze bleiben unverändert erhalten
  - − Verlagerung des Problems in die Anfragezeit . . .
- **Vermeidung**
  - Zusammenführung der Datensätze mit *Mengen-wertigen Attributen* evtl. *Maskierung*
  - − Immer noch Verlagerung des Problems in die Anfragezeit
  - − Komplexe Verwaltungslogik in der Datenquelle erforderlich
  - + Immerhin: Unterstützt Benutzer Umgang mit Konflikten

Konfidenz-basiertes Merging

- Konfidenz = Vertrauen in die Korrektheit von Datensätzen
- **Datensatzebene**
  - z.B. $R_1.c = 80\%, R_2.c = 70\% \Rightarrow$ Wähle Wert von R_1
  - Problem: ggf. Informationsverlust, z.B.
    $R_1 = \{$ Maier, Hans $\}, c = 80\%$
    $R_2 = \{$ Mayer, Hans $\}, c = 70\%$
    $R_{merged} = \{$ Maier, Hans $\}, c = 70\%$
    $\Rightarrow$ Konfidenz für Vorname verloren!
- **Optimierungen**
  - Zusätzlich alte Datensätze erhalten
    $\Rightarrow$ Noch komplexer ...
  - Konfidenzen auf Attributebene betrachten