

# Information Engineering 2

## Datenqualität & Data Matching

Prof. Dr. Kurt Stockinger

SW	Datum	Vorlesungsthema	Praktikum
1	23.02.2022	Data Warehousing Einführung	Praktikum 1: KNIME Tutorial
2	02.03.2022	Dimensionale Datenmodellierung 1	Praktikum 1: KNIME Tutorial (Vertiefung)
3	09.03.2022	Dimensionale Datenmodellierung 2	Praktikum 2: Datenmodellierung
4	16.03.2022	Datenqualität und Data Matching	Praktikum 3: Star-Schema, Bonus: Praktikum 4: Slowly Changing Dimensions
5	23.03.2022	Big Data Einführung	DWH Projekt - Teil 1
6	30.03.2022	Spark - Data Frames	DWH Projekt - Teil 2 (Abgabe: 4.4.2022 23:59:59)
7	06.04.2022	Data Storage: Hadoop Distributed File System & Parquet	Praktikum 1: Data Frames
8	13.04.2022	Query Optimization	Praktikum 2: Data Storage
9	20.04.2022	Spark Best Practices & Applications	Praktikum 3: Query Optimization & Performance Analysis
10	27.04.2022	Machine Learning mit Spark 1	Praktikum 3: Query Optimization & Performance Analysis (Vertiefung)
11	04.05.2022	Machine Learning mit Spark 2 + Q&A	Praktikum 4: Machine Learning (Regression)
12	11.05.2022	NoSQL Systems	Big Data Projekt - Teil 1
13	18.05.2022	Keine Vorlesung (Arbeit am Projekt)	Big Data Projekt - Teil 2
14	25.05.2022	Keine Vorlesung (Arbeit am Projekt)	Big Data Projekt - Teil 3 (Abgabe: 30.5.2022 23:59:59)

# Ziele der heutigen Lektionen

- Kennen von
  - unterschiedlichen Fehlerquellen in Daten
- Verstehen von Methoden zur **Duplikaterkennung**:
  - Masken und Wildecards
  - String-Distanzen
  - Phonetische Suche
- Selbststudium von Methoden zu **Data Matching (Entity Matching)**:

# Die wesentlichen betrieblichen Probleme in einem Data Warehouse



**Was ist teuer?**

Meyer, Arnold  
Meier, Noldi  
Josic, Petar  
Jositsch, Peter  
Bodino, Gerald  
Beaudinot, Gérold  
Baudinot, Gerold

**Fehler sind teuer**

**Weil Korrekturen  
sehr oft Einsatz von  
menschlicher  
Arbeitskraft  
bedeutet**

Bienne  
Biel  
Sitten  
Sion  
.....

# Woher kommen fehlerhafte Daten?

## Fehlerbeispiele:

- Duplikate
- Out of range
- Missing values
- Typos

## Ursachen:

- Ausfall Quellsystem (geplant oder Absturz)
- Manuelle Eingabe
- Fehlerhafte Konfigurationen
- Unvollständiger / asynchroner Informationsfluss
- Fehlerhafte Konvertierung:
  - z.B. PDF → Text oder Excel → CSV

# Datenqualität Teil 1: Fehler in Texten

# Beispiele für Datenqualitätsprobleme

Tippfehler	Chinois on Main	2709 Main Street	Santa Monica
	Chniois on Main	2909 Main Street	Santa Monca
Abkürzungen	Foobar Holding	Flurstrasse 10	Oetwil an der Limmat
	Foobar Hldg	Flurstr. 10	Oetwil a. d. Limmat
Abweichende Bezeichnungen	Four Seasons	854 Seventh Avenue	New York City
	4 Seasons Grill Room	854 7th Ave. between 54th and 55th Sts.	New York
Untergeordnete Ortsbezeichnungen	Grill on the Alley	9560 Dayton Way	Los Angeles
	Grill on the Alley	9560 Dayton Way	Beverly Hills
Zweisprachigkeit	Déborah François	Quellgasse 4	Biel
	Déborah François	Rue de la Source 4	Bienne
Unvollständige Angaben	Grill on the Alley	9560 Dayton Way	Los Angeles
	Grill on the Alley	(null)	Beverly Hills
Vertauschte Werte	Thomas	Blake	Santa Monica
	Blake	Thomas	Santa Monica

# Beispiele für Qualitätsprobleme: Duplikate

Variante 1	Variante 2	Variante 3	Variante 4
Meier	Meyer	Maier	
Bumann	Baumann	.....	
Baudinot	Baudino	Bodino	Beaudineau
Jositsch	Josić		
...			

- Welches Ergebnis liefert folgende SQL-Query?  
select distinct Name from TableX
- Wie korrigiert man solche Fehler?





# Methoden für Duplikatserkennung

1. Masken / Wildcards:
  - Erkennen von Mustern
2. String-Distanzen:
  - Vergleiche Buchstaben und berechne Anzahl unterschiedlicher Zeichen
3. Phonetische Suche:
  - Ähnlich klingende Namen werden mit ähnlichem Code abgebildet

# 1) Masken / Wildcards

Suche mit **Masken / Wildcards** → Beschreibung von Mustern im Text

Variante 1	Variante 2	Maske	Verfahren
Meier	Meier		
Meier	Meyer	Me?er	Regex
Meier	Maier	M?ier	Regex
Bumann	Baumann	B.*umann	Regex
Baudinot	Baudino	Baudino.*	Regex
Baudinot	Bodino		?
Baudinot	Beaudineau		?
Jositsch	Josić		?

# Vor- und Nachteile

- Vorteil:
  - Sehr einfach anwendbar für konkrete Fragestellungen
- Nachteil:
  - Suchmuster muss genau bekannt sein

## 2) String-Distanz: Levenshtein Distanz

- Verfahren eignet sich zum Bestimmen möglicher Matches
- Gilt auch als „**Editierdistanz**“
- Prinzip:
  - Wie viele String-Operationen müssen ausgeführt werden, damit die Strings übereinstimmen?
- Operationen:
  - Ersetzung
  - Löschung
  - Einfügung

### Beispiel:

- Levenshtein Distanz(Ger**o**ld, Ger**a**ld) = 1
- Levenshtein Distanz(Hilde**g**ard, Hilde) = 4

# Beispiele: Levenshtein

- Meier vs. Mayer
- Tsar vs. Zar
- Deighton vs. Dayton



# String-Distanzen: Jaro-Winkler Distanz

- Misst Distanz zweier Strings A und B
- Normalisiertes Ähnlichkeitsmass:
  - 0 ... Keine Ähnlichkeit
  - 1... Strings sind identisch

$$d_{jaro}(A, B) = \frac{1}{3} \cdot \left( \frac{m}{|A|} + \frac{m}{|B|} + \frac{m-t}{m} \right) \text{ für } m > 0, \quad d_{jaro} = 0 \text{ für } m = 0$$

**m... Matching:** Anzahl der übereinstimmenden Zeichen in A und B, falls Abstand nicht grösser als

$$\text{floor} \left( \frac{\max(|A|, |B|)}{2} \right) - 1$$

**t ...** Halbe Anzahl der darin notwendigen **Transpositionen**

- Transposition:
  - Jedes Zeichen des Strings A wird mit jedem Zeichen des Strings B verglichen
  - Die Anzahl der Operationen, um Matches zu erhalten, geteilt durch 2 entspricht der Anzahl der Transpositionen

- Beispiel:
  - String A: CRATE
  - String B: TRACE

$$\text{floor} \left( \frac{\max(|A|, |B|)}{2} \right) - 1$$

- A) Matches: R, A, E
  - $m = 3$
- B) Keine Matches: C, T
  - Warum?



# Jaro-Winkler Beispiel

$$d_{jaro}(A, B) = \frac{1}{3} \cdot \left( \frac{m}{|A|} + \frac{m}{|B|} + \frac{m-t}{m} \right) \text{ für } m > 0, \quad d_{jaro} = 0 \text{ für } m = 0$$

$d_{jaro}(\text{Winkler}, \text{Winkel}) = ?$

$m = ?$

$t = ?$





# Lösung

$$m=6$$

Abstand muss kleiner als  $\text{floor}(7/2) = 3 - 1 = 2$  sein

$t=2/2$  („L“ in Winkler  $\rightarrow$  löschen, R  $\rightarrow$  L: „R“ in Winkler durch „L“  
ersetzen) ... T = halbe Anzahl der notwendigen Transpositionen

$$1/3 * (6/7 + 6/6 + (6-1)/6) = 0.897$$

- Vorteile:
  - Sehr generell einsetzbar
  - Funktioniert gut bei „kleinen Tippfehlern“
  - Jaro-Winkler ist echtes Ähnlichkeitsmass (0 bis 1)
- Nachteile:
  - Funktioniert nur bei kurzen Strings (Jaro-Winkler)

# 3) Phonetische Suche

- **Phonetische Suche** in Texten
  - Beispiel: Mayer und Meier tönen ähnlich, Bodino und Baudinot auch
- ➔ **Soundex** ([www.sound-ex.de](http://www.sound-ex.de) , patentiert durch Russel & Odell, 1918)
  - phonetischer Algorithmus zur Indizierung von Wörtern und Phrasen nach ihrem Klang
  - Soundex Code besteht aus einem Buchstaben (Anfangsbuchstabe) gefolgt von drei Ziffern
- **Das Verfahren:**
  1. Erster Buchstabe ist Teil des Codes
  2. Vokale, Umlaute und H, W, Y entfernen
  3. Restliche Buchstaben anhand Tabelle umkodieren
  4. Max. 3 Kodierungen
- **Beispiel:**  
**Baudinot, Boudinot, Beaudinot** → B353

Buchstaben	Code
B, F, P, V	1
C, G, J, K, Q, S, X, Z	2
D, T	3
L	4
M, N	5
R	6

# Beispiele von Soundex

- Meier vs. Mayer
- Tsar vs. Zar
- Deighton vs. Dayton
- Hoffmann vs. Heppenheimer



Buchstaben	Code
B, F, P, V	1
C, G, J, K, Q, S, X, Z	2
D, T	3
L	4
M, N	5
R	6

# Lösung

M600, M600

T260, Z600

D235, D350

H155, H155

# Nachteil von Soundex

- Stark abhängig vom ersten Buchstaben
- Funktioniert gut für Englisch aber nicht für Deutsch

# Kölner Phonetik

- Spezielle Anpassung an deutsche Sprache
- Mächtigere Ersetzungstabelle
- Auch erster Buchstabe wird kodiert
- Wort wird vollständig kodiert, d.h. nicht nur 3 Ziffern
- Umwandlung erfolgt in **3 Schritten**:
  1. Buchstabenweise Kodierung von links nach rechts entsprechend der Umwandlungstabelle.
  2. Entfernen aller mehrfach nebeneinander vorkommenden Ziffern.
  3. Entfernen aller Codes "0" ausser am Anfang.

# Beispiel Kölner Phonetik

Eingangsbuchstabe	Kontext	Code
A,E,I,J,O,U,Y		0
H		-
B		1
P	nicht vor H	1
D,T	nicht vor C,S,Z	2
F,V,W		3
P	vor H	3
G,K,Q		4
C	im Anlaut vor A,H,K,L,O,Q,R,U,X	4
C	vor A,H,K,O,Q,U,X ausser nach S,Z	4
X	nicht nach C,K,Q	48
L		5
M,N		6
R		7
S,Z		8
C	nach S,Z	8
C	im Anlaut ausser vor A,H,K,L,O,Q,R,U,X	8
C	nicht vor A,H,K,O,Q,U,X	8
D,T	vor C,S,Z	8
X	nach C,K,Q	8

Tsar vs. Zar

Hoffmann vs. Heppenheimer





Tsar 8807 => 87 vs. Zar: 807 => 87 (gleich)

Hoffmann: 0366, Heppenheimer: 01667 (ungleich)

# Vor- und Nachteile phonetischer Codes

- Vorteile:
  - Sehr gut für Namensabgleichung geeignet
  - Soundex für Englisch optimiert
  - Kölner Phonetik für Deutsch optimiert
- Nachteile:
  - Sprachabhängig
  - Kleine Tippfehler können grosse Änderungen bewirken
  - Kein richtiges Ähnlichkeitsverfahren (alles oder nichts)

# Alternative Lösungen


- Welche Möglichkeiten gibt es noch?



# Wie hilft das bei der Duplikatseliminierung?

- Phonetische Funktionen erzeugen **Hash Codes**
- Bei Hash Codes gibt es immer Kollisionen
- Bei Duplikatseliminierung sind **Kollisionen erwünscht**:

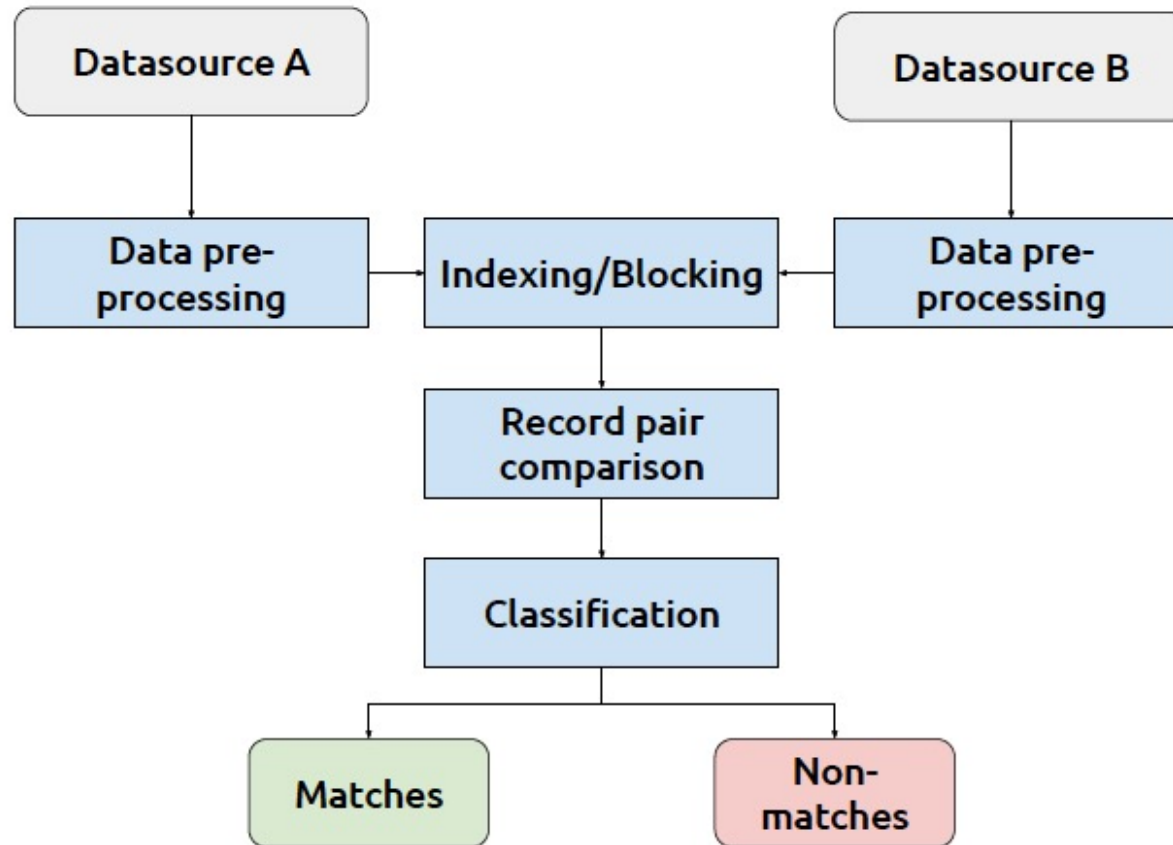
Name	Hash
Baumann	B550
Bumann	B550
Buman	B550



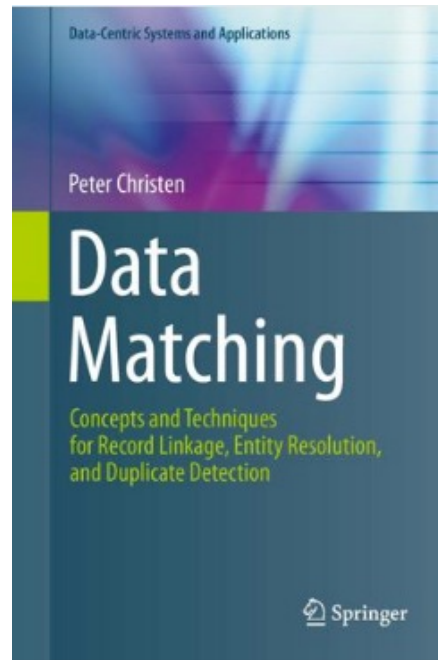
Kollision

➔ Nachfolgende Anwendung von SELECT DISTINCT

# Data Matching Problem (Entity Matching)



# Literatur für Data Matching



<https://link.springer.com/book/10.1007%2F978-3-642-31164-2>

The Data Matching Process, S. 23-35

## Deep Learning for Entity Matching: A Design Space Exploration

Sidharth Mudgal<sup>1</sup>, Han Li<sup>1</sup>, Theodoros Rekatsinas<sup>1</sup>, AnHai Doan<sup>1</sup>,  
Youngchoon Park<sup>2</sup>, Ganesh Krishnan<sup>3</sup>, Rohit Deep<sup>3</sup>, Esteban Arcaute<sup>4</sup>, Vijay Raghavendra<sup>3</sup>

<sup>1</sup>University of Wisconsin-Madison, <sup>2</sup>Johnson Controls, <sup>3</sup>@WalmartLabs, <sup>4</sup>Facebook

### ABSTRACT

Entity matching (EM) finds data instances that refer to the same real-world entity. In this paper we examine applying deep learning (DL) to EM, to understand DL's benefits and limitations. We review many DL solutions that have been developed for related matching tasks in text processing (e.g., entity linking, textual entailment, etc.). We categorize these solutions and define a space of DL solutions for EM, as embodied by four solutions with varying representational power: SIF, RNN, Attention, and Hybrid. Next, we investigate the types of EM problems for which DL can be helpful. We consider three such problem types, which match structured data instances, textual instances, and dirty instances, respectively. We empirically compare the above four DL solutions with Magellan, a state-of-the-art learning-based EM solution. The results show that DL does not outperform current solutions on structured EM, but it can significantly outperform them on textual and dirty EM. For practitioners, this suggests that they should seriously consider using DL for textual and dirty EM problems. Finally, we analyze DL's performance and discuss future research directions.

### KEYWORDS

Deep learning; entity matching; entity resolution

can automatically construct important features, thereby obviating the need for manual feature engineering. This has transformed fields such as image and speech processing, medical diagnosis, autonomous driving, robotics, NLP, and many others [28, 46]. Recently, DL has also gained the attention of the database research community [17, 83].

A natural question then is whether deep learning can help entity matching. Specifically, has DL been applied to EM and other related matching tasks? If so, what are those tasks, and what kinds of solutions have been proposed? How do we categorize those solutions? How would those DL solutions compare to existing (non-DL) EM solutions? On what kinds of EM problems would they help? And on what kinds of problems would they not? What are the opportunities and challenges in applying DL to EM? As far as we know, no published work has studied these questions in depth.

In this paper we study the above questions, with the goal of understanding the benefits and limitations of DL when applied to EM problems. Clearly, DL and EM can be studied in many different settings. In this paper, as a first step, we consider the classic setting in which we can automatically train DL and EM solutions on *labeled training data*, then apply them to test data. This setting excludes unsupervised EM approaches such as clustering, and approaches that require substantial human effort such as crowdsourced EM or

<http://pages.cs.wisc.edu/~anhai/papers1/deepmatcher-sigmod18.pdf>

# ZHAW-Forschung an Entity Matching

## Entity Matching on Unstructured Data: An Active Learning Approach

Ursin Brunner and Kurt Stockinger  
ZHAW Zurich University of Applied Sciences, Switzerland

**Abstract**—With the growing number of data sources in enterprises, *entity matching* becomes a crucial part of every data integration project. In order to reduce the human effort involved in identifying matching entities between different database tables, typically machine learning algorithms are applied. Moreover, active learning is often combined with supervised machine learning methods to further reduce the effort of labeling entities as true or false matches. However, while state-of-the-art active learning algorithms have proven to work well on structured data sets, unstructured data still poses a challenge in entity matching.

This paper proposes an *end-to-end entity matching pipeline* to minimize the human labeling effort for entity matching on unstructured data sets. We use several natural language processing techniques such as *soft tf-idf* to pre-process the record pairs before we classify them using a novel *Active Learning with Uncertainty Sampling (ALWUS)* algorithm. We designed our algorithm as a plugin system to work with any state-of-the-art classifier such as support vector machines, random forests or deep neural networks. Detailed experimental results demonstrate that our *end-to-end entity matching pipeline* clearly outperforms comparable entity matching approaches on an unstructured real-world data set. Our approach achieves significantly better scores (F1-score) while using 1 to 2 orders of magnitude fewer human labeling efforts than existing state-of-the-art algorithms.

matching process as a classification problem where we need to minimize the number of false positives.

TABLE I  
DATABASE A

Surname	GivenName	Street	City
Meyer	Marie	3/12-14 Hope Cnr	Sydney
Smith	John	42 Miller St	Canberra

TABLE II  
DATABASE B

Name	Address
Meier, Mary	14 (App 3) Hope Corner, Sydney 2000
Jonny Smith	47 Miller Street, 2619 Canberra ACT

While early entity matching was heavily used in the health sector and in national censuses [5], it is now a challenge that appears in numerous application domains. As large companies produce (and consume) more and more data which originates from multiple data sources, the process of *data integration*

## Entity Matching with Transformer Architectures - A Step Forward in Data Integration

Ursin Brunner  
Zurich University of Applied Sciences  
Switzerland  
ursin.brunner@zhaw.ch

Kurt Stockinger  
Zurich University of Applied Sciences  
Switzerland  
kurt.stockinger@zhaw.ch

### ABSTRACT

Transformer architectures have proven to be very effective and provide state-of-the-art results in many natural language tasks. The attention-based architecture in combination with pre-training on large amounts of text lead to the recent breakthrough and a variety of slightly different implementations.

In this paper we analyze how well four of the most recent attention-based transformer architectures (BERT[6], XLNet[33], RoBERTa[17] and DistilBERT [23]) perform on the task of entity matching - a crucial part of data integration. Entity matching (EM) is the task of finding data instances that refer to the same real-world entity. It is a challenging task if the data instances consist of long textual data or if the data instances are "dirty" due to misplaced values.

To evaluate the capability of transformer architectures and transfer-learning on the task of EM, we empirically compare the four approaches on inherently difficult data sets. We show that transformer architectures outperform classical deep learning methods in EM[7, 20] by an average margin of 27.5%.

or require large efforts in hand-crafted features [2, 13]. Therefore and due to the recent advances of deep learning in natural language processing (NLP), several papers suggest end-to-end deep learning architectures [7, 20, 34] for EM.

Table 1: Database A - structured product information, with *description* being a text-blob.

Title	Brand	Description	Price
iPhone XS	Apple	The brand new iPhone now available in white, red and silver.	899.99
ZenFone 4 Pro	Asus	Thin and light, yet incredibly strong, the ZenFone 4 Pro (ZS551KL) features an expansive 5.5-inch, Full HD AMOLED display	530.00

Swiss Data Science Conference, 2019

International Conference on  
Extending Database Technology, 2020



# Zusammenfassung

- Datenqualitätsprobleme werden oft **unterschätzt**
- Thema ist fast für jedes Unternehmen von **höchster Relevanz**
- Leider wird das Thema oft als **"langweilig"** empfunden
- Mit **Maschine Learning** lassen sich viele interessante Datenqualitätsprobleme lösen und machen das Thema wieder **"spannend"**