Willkommen!

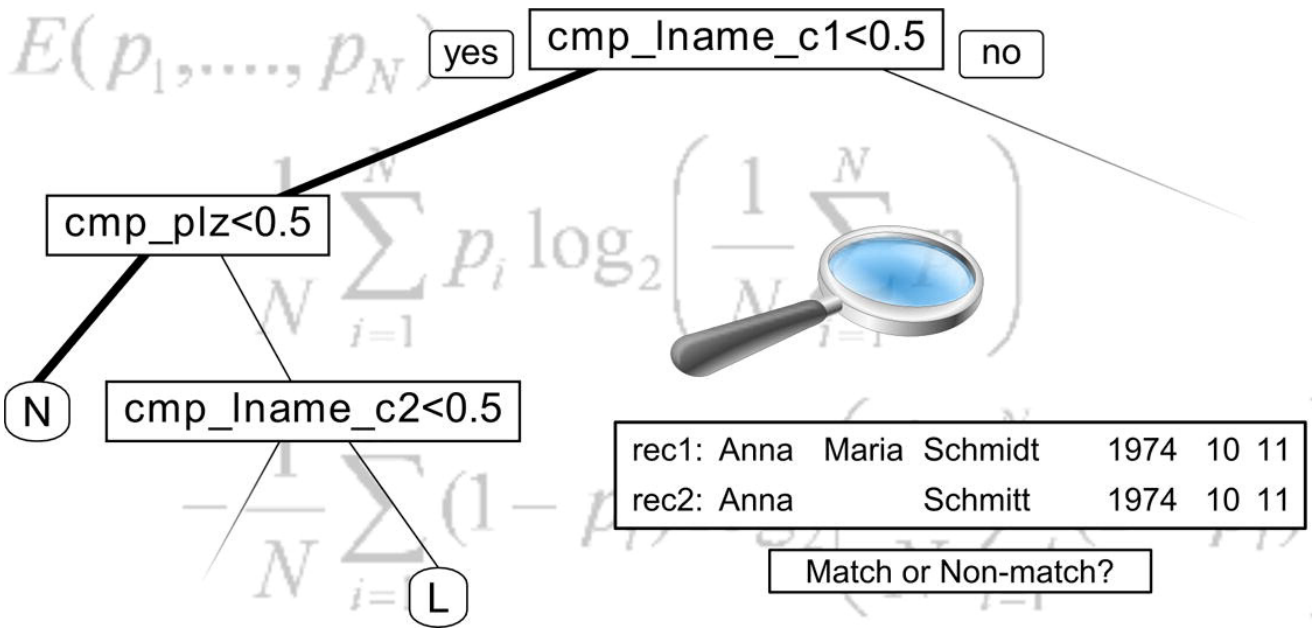
The Relevance and Hands-on Application of Biomedical Record Linkage in the Big Data Era

Prof. Dr. Murat Sariyar

- Berner Fachhochschule TI

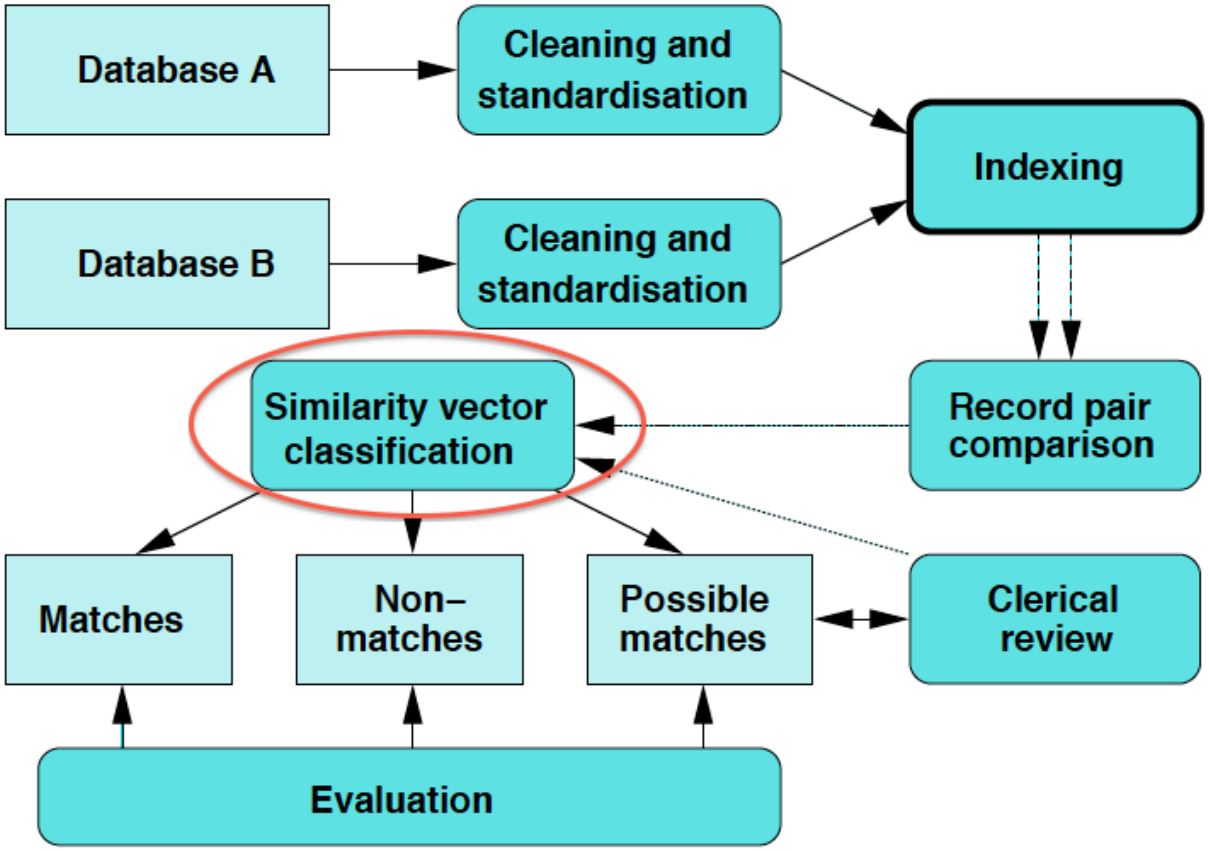
Introduction to Record Linkage

What is Record Linkage?



https://ars.els-cdn.com/content/image/1-s2.0-S1532046412000238-fx1_lrg.jpg

What is Record Linkage?



Our research on Record Linkage

Our methodological research in Record Linkage:

- Various machine Learning methods for finding duplicates
- Extreme value statistics for determining thresholds in statistical RL models
- Dealing with missing values in the context of RL
- Active Learning for determining training sets for RL problems

We involve students in RL projects

Our preliminary conceptual research in Record Linkage:

- Operationalizing qualitative identity (versus numerical identity)
- Relevance of identity criteria (sortals) for understanding identity in RL contexts
- Different forms of explanations for ML results (Luhmann and Bokulich)

Practical use cases

Linking data in a cancer registry and a register of residents:

- linked based on demographic attributes to track the survival of patients
- Probabilistic Record Linkage with two threshold:

('Urs', 'Schmidt', 'Bern', '18', '11', '1990', 'm') -- data from the register of residents

('Urz', 'Schmitt', 'Berne', '18', '11', '1990', 'm') -- cancer registry data

γ=(1, 0.86, 0.8, 1, 1, 1, 1) -- use a string metric

$$\mathrm{w} = \log\left(\frac{(P(\gamma|M)}{(P(\gamma|U)}\right)$$ -- Probs estimated in a Fellegi-Sunter model

Development of the RL procedure in the context of the method test for the register census at the German Federal Statistical Office:

- Using Apache Spark and Splink
- Applying active learning and probabilistic Record Linkage
- Using Association rules and NLP for finding blocking rules

Why conceptual research?

We usually rely on certain demographic attributes and their similarity, yes, but there are two central problems

(i) we want to differentiate between changes in the same entity (e.g., the same virus exhibits new characteristics) and differences that relate to different entities (e.g., a new virus has emerged) and

(ii) there is a lack of guidance on how to resolve synonyms (false non-matches) and homonyms (false matches), especially in the training phase of the algorithms.

For both problems, similarity is a too simple concept.

Software for RL

Our RecordLinkage R package on CRAN

- Provides functions for linking and deduplicating data sets.
- Over 279'000 downloads since 2020-04-09
- Methods based on a stochastic approach are implemented as well as classification algorithms from the machine learning domain.
- Can deal with high-volume data by using the ff package
- However, it requires an Apache-Spark adaptation to be able to efficiently process the data volume at the the German Federal Statistical Office
- For details, see "The RecordLinkage Package: Detecting Errors in Data"

The Python package Splink

Splink is a PySpark package that implements the Fellegi-Sunter model of record linking and enables parameters to be estimated using the EM algorithm

- Comprehensive graphical output showing parameter estimates and iteration history make it easier to understand the model and diagnose convergence
- Support for deduplication, linking, and a combination of both, including support for deduplicating and linking multiple data sets.
- Greater customizability of record comparisons, including the ability to specify custom, user defined comparison functions.
- Term frequency adjustments on any number of columns.
- It's possible to save a model once it's been estimated - enabling a model to be estimated, quality assured, and then reused as new data becomes available.

What is Apache Spark?

Computing engine for clusters (developed at UC Berkeley)

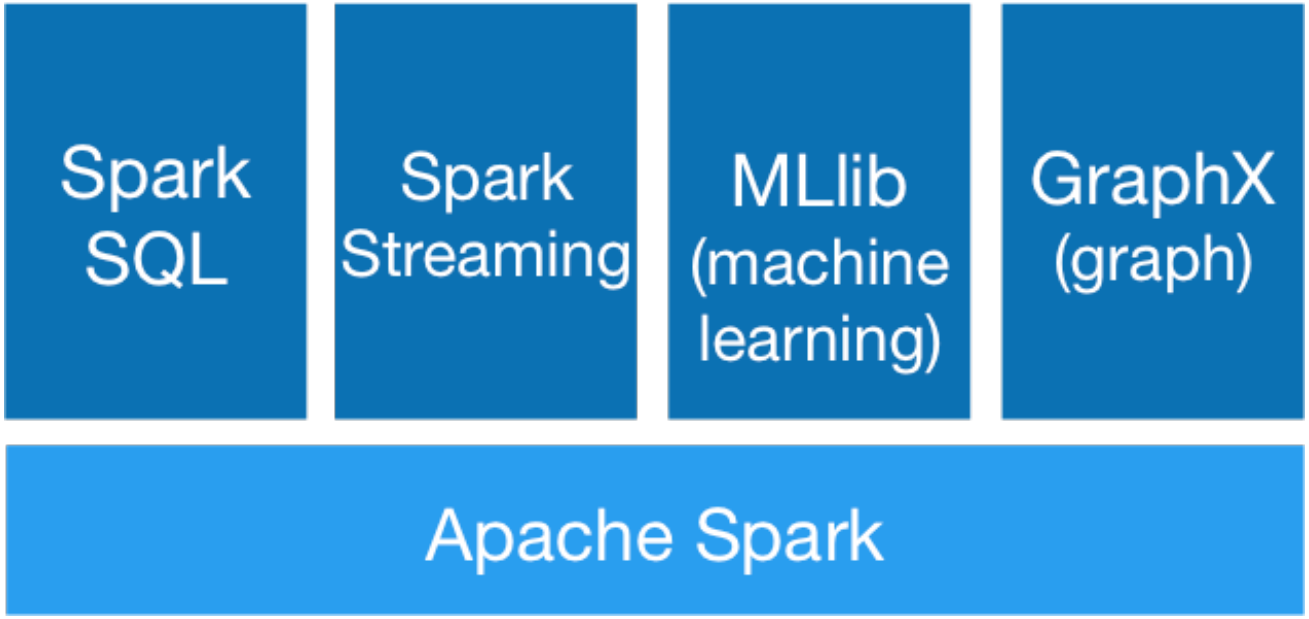
- Allows large amounts of data to be processed (GB-PB-TB)
- Supports Java, Scala, Python and R
- Libraries for SQL, ML and graph processing
- 10-100x faster than Hadoop Map/Reduce
- So, Spark CAN a Hadoop enhancement to MapReduce.

Programming model

- Computing of distributed datasets (RDDs with Spark Dataframe on top).
- Automatic error correction
- Parallel processing (map, filter, ...)
- Data can be in RAM or on the disks of the cluster

In Splink without hadoop implementation: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

What is Apache Spark?



What is Apache Spark?

```
>>> import pyspark.pandas as ps
>>>
>>> psdf = ps.range(10)
>>> pdf = psdf.to_pandas()
>>> pdf.values
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5],
       [6],
       [7],
       [8],
       [9]])
```

```
>>> import pyspark.pandas as ps
>>>
>>> psdf = ps.range(10)
>>> sdf = psdf.to_spark().filter("id > 5")
>>> sdf.show()
+---+
| id|
+---+
|  6|
|  7|
|  8|
|  9|
+---+
```

https://spark.apache.org/docs/latest/api/python/user_guide/pandas_on_spark/pandas_pyspark.html

Software demonstration

Jupyter Notebooks

9 Notebooks for the German Federal Statistical Office were developed:

- 1-Splink-Python-Intro.ipynb: Intro into Python and DuckDB
- 2-Splink-Einfach-PySpark.ipynb: Deduplication with PySpark
- 3-Splink-Einfach-DBDuck.ipynb: Deduplication with DuckDB
- 4-Splink-SQL.ipynb: SQL extensions for Splink
- 5-Splink-Linkage1.ipynb: Linkage with DuckDB
- 6-Splink-Linkage2.ipynb: Linkage with PySpark and Threshold determination
- 7-Splink-ActiveL.ipynb: Informative-based active learning
- 8-Splink-AssocRules.ipynb: association rules for det. matching & blocking
- 9-Splink-BERT.ipynb: NLP for finding blocking rules and valid names

Screenshots - DuckDB backend

```python
# from splink.spark.spark_linker import SparkLinker
from splink.duckdb.duckdb_linker import DuckDBLinker

basic_settings = {
    "unique_id_column_name": "rec_id",
    "link_type": "link_only",
    # NB as we are linking one-one, we know the probability that a random pair will be a match
    # hence we could set:
    # "probability_two_random_records_match": 1/5000,
    # however we will not specify this here, as we will use this as a check that
    # our estimation procedure returns something sensible
}

# linker = SparkLinker(df, settings)
linker = DuckDBLinker(dfs, basic_settings)
```

Screenshots - PySpark backend

```python
from splink.spark.jar_location import similarity_jar_location

from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from pyspark.sql import types

conf = SparkConf()
# This parallelism setting is only suitable for a small toy example
conf.set("spark.driver.memory", "12g")
conf.set("spark.default.parallelism", "16")


# Add custom similarity functions, which are bundled with Splink
# documented here: https://github.com/moj-analytical-services/splink_scalaudfs
path = similarity_jar_location()
conf.set("spark.jars", path)

sc = SparkContext.getOrCreate(conf=conf)

spark = SparkSession(sc)
spark.sparkContext.setCheckpointDir("./tmp_checkpoints")

# Register the jaro winkler custom udf
spark.udf.registerJavaFunction(
    "jaro_winkler", "uk.gov.moj.dash.linkage.JaroWinklerSimilarity", types.DoubleType()
)
```

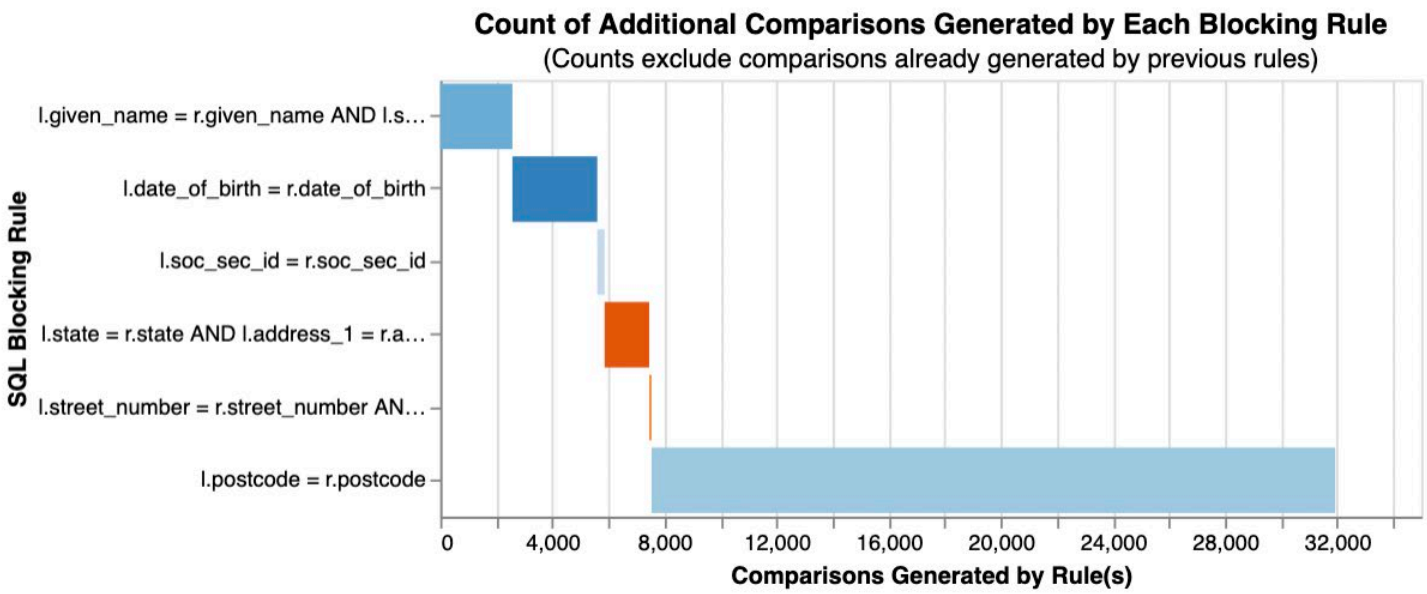
Screenshots - Record pairs

	rec_id	given_name	surname	street_number	address_1	address_2	suburb	postcode	state	date_of_birth	soc_sec_id	cluster
0	rec-1070-org	michaela	neumann	8.0	stanley street	miami	winston hills	4223	nsw	19151111	5304218	rec-1070
1	rec-1016-org	courtney	painter	12.0	pinkerton circuit	bega flats	richlands	4560	vic	19161214	4066625	rec-1016

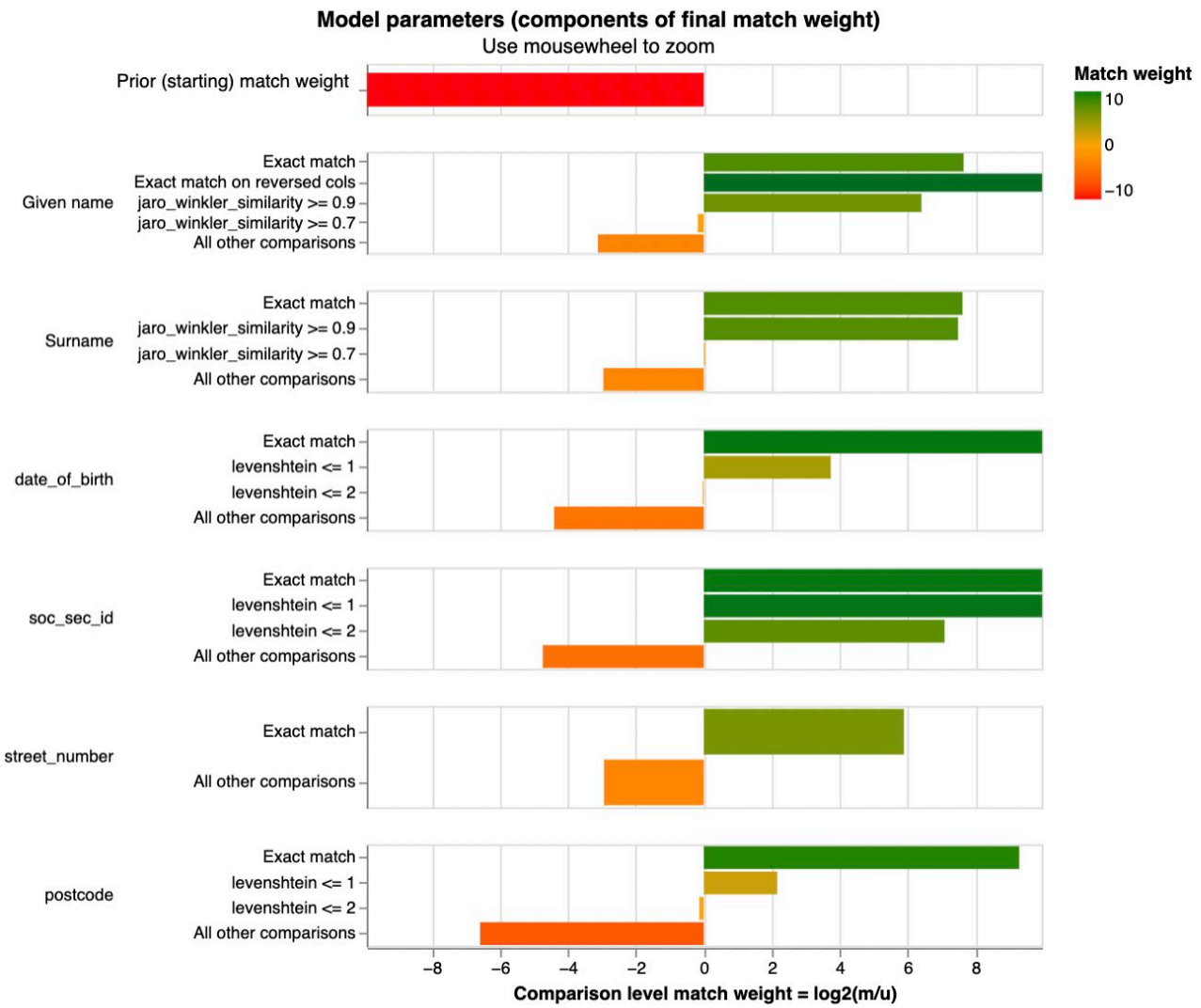
	rec_id	given_name	surname	street_number	address_1	address_2	suburb	postcode	state	date_of_birth	soc_sec_id	cluster
0	rec-561-dup-0	elton	NaN	3.0	light setreet	pinehill	windermere	3212	vic	19651013	1551941	rec-561
1	rec-2642-dup-0	mitchell	maxon	47.0	edkins street	lochaoair	north ryde	3355	nsw	19390212	8859999	rec-2642

Screenshots - Blocking rules

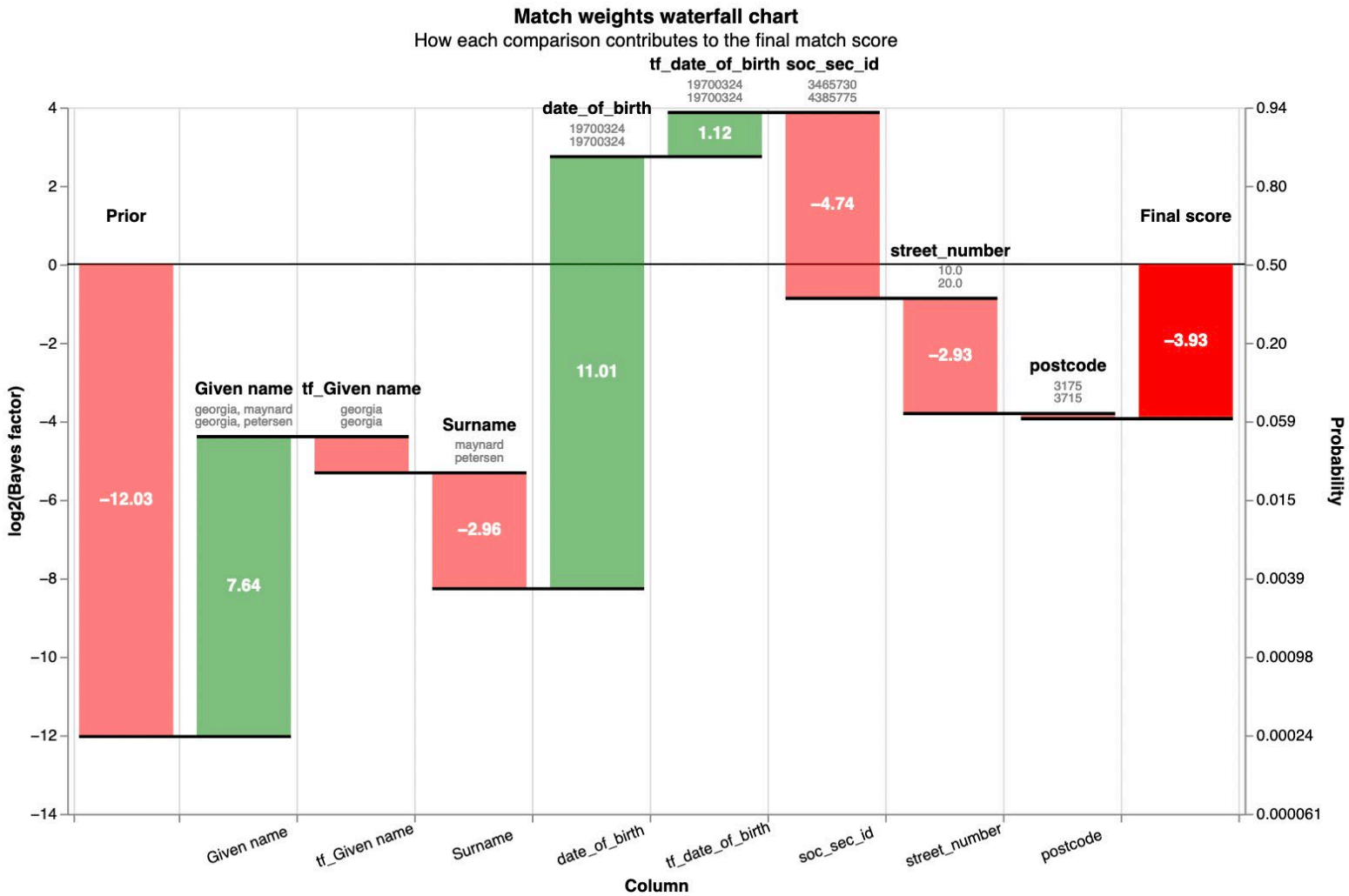
```
blocking_rules = [
    "l.given_name = r.given_name AND l.surname = r.surname",
    "l.date_of_birth = r.date_of_birth",
    "l.soc_sec_id = r.soc_sec_id",
    "l.state = r.state AND l.address_1 = r.address_1",
    "l.street_number = r.street_number AND l.address_1 = r.address_1",
    "l.postcode = r.postcode",
]
linker.cumulative_num_comparisons_from_blocking_rules_chart(blocking_rules)
```



Screenshots - Match weight chart



Screenshots - Match weight waterfall chart



Conceptual research

What is identity? The legal view

GDPR Article 4:

"… an *identifiable* person is one who can be *identified* directly or indirectly by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social **identity**."

⇒ Looks somehow circular
⇒ Philosophy might help but on which level?
⇒ Metaphysics of identity has a revival since the realist turn in philosophy

Salmon on identity

"For surely, there is no qualitative fact about x, other than the fact of its possible experience, in virtue of which x=x.

It follows by Leibniz's law that if x=y, then y is also such that there is nothing in its qualitative character, that makes x=y.

The point is not that one cannot reasonably expect to reasonably to find a criterion of identity in many cases (a cautionary point made long ago by Kripke) but that the quest for such a criterion is misguided in principle."

⇒ It seems like a dead end!

⇒ Metaphysics does not help, but epistemology might do

Move from metaphysics to epistemology

Two important distinctions:

- **Numerical identity**: the same entity is referenced at different times, even though it might be denoted differently (It is always the same object?)
- **Qualitative identity**: two entities have the same properties (they are indiscernible) but are not necessarily one and the same.

The definition of numerical identity provides no guidance for the practice, since it remains unclear on what basis the different signifiers (labels or words) are defined as referring to the same entity, it is just (trivial) ontology

⇒ Use qualitative identity for operationalization: relational identity

Relational identity for identification

Listing of all relations that an entity x has within itself (its attributes) and to other entities (https://iep.utm.edu/differential-ontology/).

Relational similarity: compares the relations of concepts (A::B) with other ones (C::D). Values for such relationships can be generated by word embeddings

Example: the name of a patient is not just an attribute of the patient, but a relation between the patient and those authorities that have certified and validated that name assignment. Hence, relations such as "identified by" or "certified by" are part of the relational identity of the patient.

Relational word embeddings (left side)

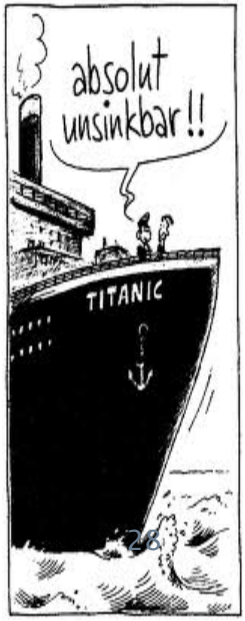
INNOCENT-NAIVE	
RWE	FastText
vain-selfish	murder-young
honest-hearted	imprisonment-term
cruel-selfish	conspiracy-minded
SHOCK-GRIEF	
RWE	FastText
anger-despair	overcome-sorrow
anger-sorrow	overcome-despair
anger-sadness	moment-sadness

https://core.ac.uk/download/pdf/227034256.pdf

It's your turn:

(Question)

Should we really declare "There is nothing to identity" (numerical identity) on the ontological level and focus on the epistemological problems (qualitative identity)? Maybe, there is more to farm in metaphysics.



Nächste Seminare

Biel / Bienne
Quellgasse 21, Aula

1.6.2023 | Averaging Model for Feedback Control of Ultrasonic Transducers Diego Stutzer, Institute for Human Centered Engineering HuCE, BFH-TI

15.6.2023 | Intégration d'un ensemble complet de logiciels pour la conduite autonome Ahmed Hanachi, Institut pour la recherche sur l'énergie et la mobilité IEM, BFH-TI

Burgdorf / Berthoud
Pestalozzistrasse 20, E013

25.5.2023 am Jlcoweg 1 | **What is High Voltage Engineering about?** Prof. Dr. Roman Grinberg, Institute for Energy and Mobility Research IEM, BFH-TI

8.6.2023 | Waghalsige Holzkonstruktionen unter Anwendung moderner Technologie neu denken Matias Penrroz, Institut für digitale Bau- und Holzwirtschaft IdBH, BFH-AHB