Data Integration (DINT)

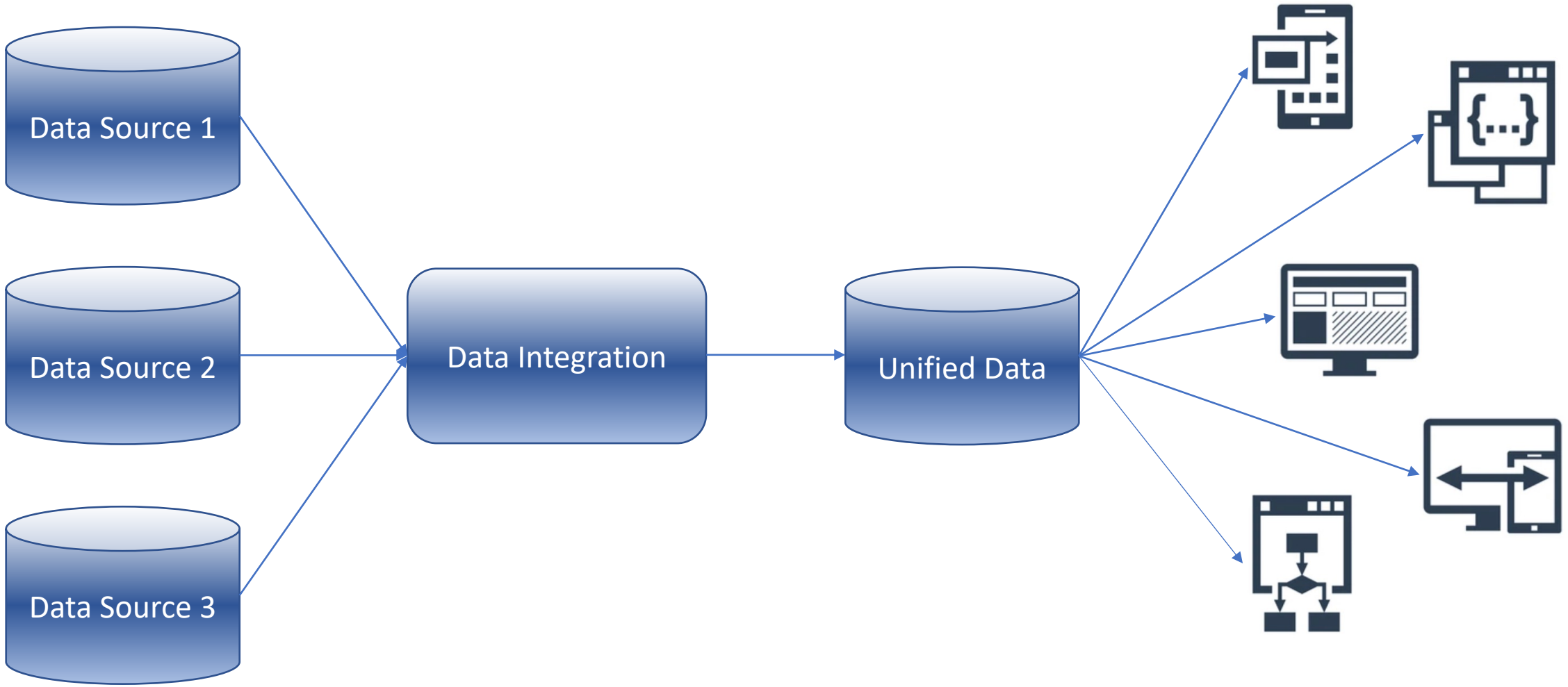
Shenghui Wang & Maurice van Keulen

17 November 2022

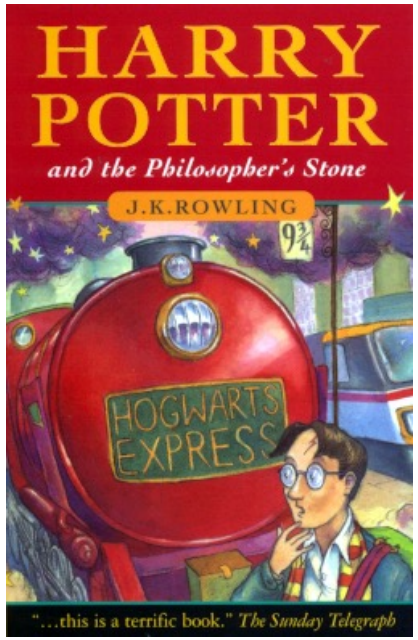
Agenda

- What is data integration?
- Data integration challenges
- Ontology matching
- Evaluation
- DINT assignment

Data Integration



Let’s look at an example



- **Title**: Harry Potter and the Philosopher’s Stone
- **Author**: J. K. Rowling
- **Illustrator**: Thomas Taylor
- **Publication year**: 1997
- **Publisher**: Bloomsbury (UK)
- **ISBN**: 0-7475-3269-9

WIKIPEDIA
The Free Encyclopedia

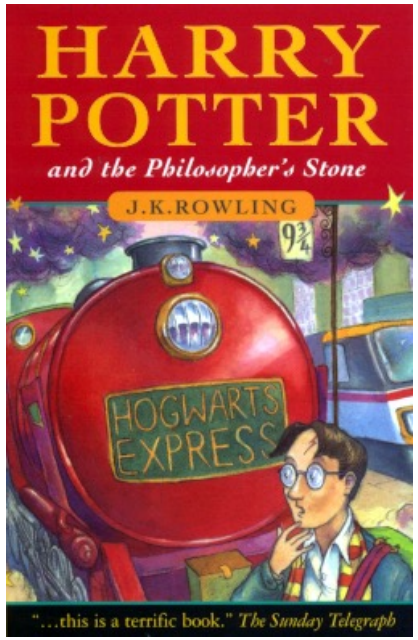
BRITISH LIBRARY

- **Title**: Harry Potter and the Philosopher’s Stone
- **Author**: J. K. Rowling
- **Subjects:** Juvenile fiction, fantasy fiction
- **Dewey**: 823.914 <- English fiction, between 1945 and 1999
- **Publication details**: London : Bloomsbury, 1997 2004 printing.
- **ISBN**: 0747574472 (pbk)

LIBRARY
LIBRARY OF CONGRESS

- **Title**: Harry Potter and the sorcerer’s stone
- **Uniform title**: Harry Potter and the philosopher’s stone
- **Personal name**: Rowling, J. K., author
- **Published/Produced**: New York, NY : Scholastic Inc., [2018]
- **ISBN**: 9781338299144 (paperback), 133829914X (paperback)
- **LC Subjects**: Wizards—Fiction, Magic-–Fiction, Schools-Fiction
- **Dewey class no.:** 823/.914 [Fic]
- **Summary**: Rescued from the outrageous neglect of his aunt and uncle, a young boy with a great destiny proves his worth while attending Hogwarts School for Witchcraft and Wizardry.

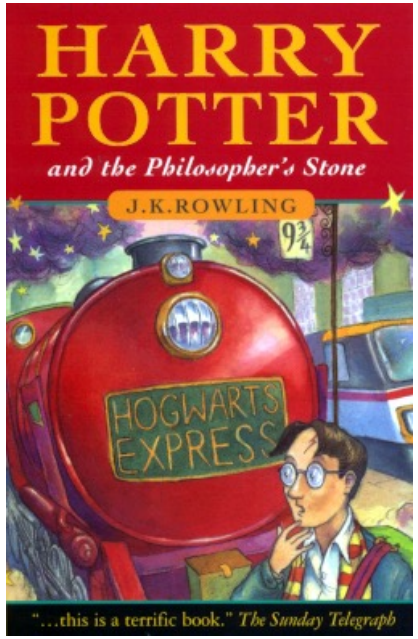
After data integration



- **Title**: Harry Potter and the Philosopher's Stone
- **Other title**: Harry Potter and the sorcerer's stone
- **Author**: J. K. Rowling
- **Illustrator**: Thomas Taylor
- **Publication year**: 1997
- **Publisher**: Bloomsbury (UK)
- **ISBN**: 0-7475-3269-9, 0747574472 (pbk), 9781338299144 (paperback), 133829914X (paperback)
- **Subjects:** Juvenile fiction, fantasy fiction, Wizards—Fiction, Magic-–Fiction, Schools-Fiction
- **Dewey**: 823.914
- **Summary**: Rescued from the outrageous neglect of his aunt and uncle, a young boy with a great destiny proves his worth while attending Hogwarts School for Witchcraft and Wizardry.

Data integration challenges

- No reliable shared IDs -> entities need to be **matched**
  - One entity has multiple candidate matches -> how to select the best one

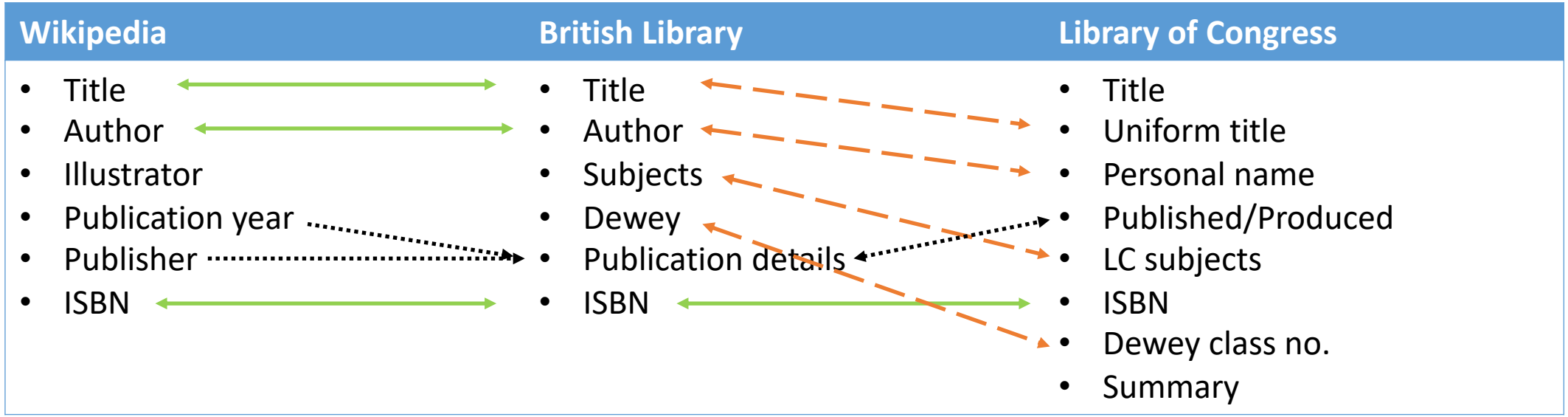


Search results from [WorldCat.org]

1.	Harry Potter and the philosopher's stone by J K Rowling	Print book : Fiction : Juvenile audience	English	1997	London : Bloomsbury Publishing Plc
2.	Harry Potter and the philosopher's stone by J K Rowling	Print book : Elementary and junior high school : Fiction	English	1997 1st British ed	London : Bloomsbury
3.	Harry Potter and the philosopher's stone by J K Rowling	Print book : Fiction : Juvenile audience	English	1997	London Bloomsbury
4.	Harry Potter y la piedra filosofal = Harry Potter and the sorcerer's stone / translated by Alicia Dellepiane Rawson by J K Rowling; Mary GrandPrÈe	Print book : Fiction : Juvenile audience	English	1997	Barcelona : Ediciones Salamandra
5.	Harry Potter and the sorcerer's stone by J K Rowling; Mary GrandPré	Print book : Elementary and junior high school : Fiction	English	1997	New York : Scholastic
6.	Harry Potter and the Philosopher's stone : [Chinese language]. by J K Rowling	Print book : Fiction	English	1997	Taipei : Lian jing

Data integration challenges

- Attributes may have different names-> entity attributes need to be **mapped**



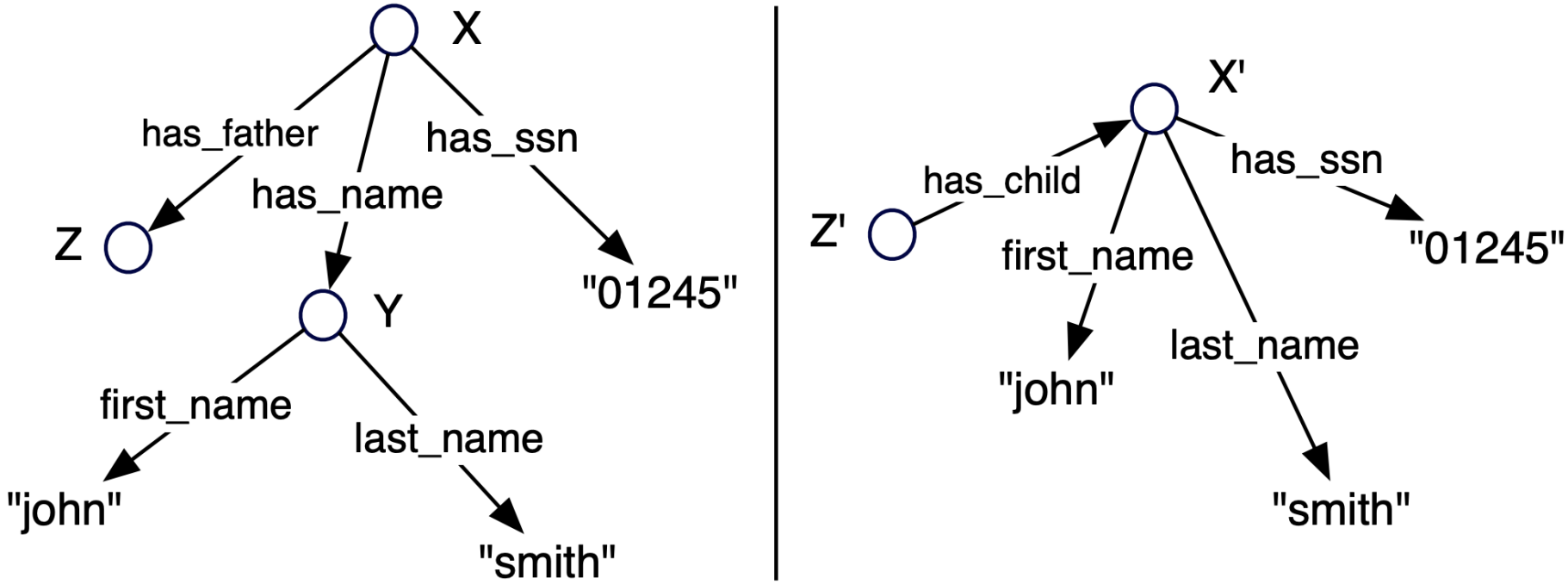
Data integration challenges

- Attribute values are not the same -> inconsistency needs to be resolved during **merging**

	Wikipedia	British Library	Library of Congress
Title	Harry Potter and the Philosopher's Stone	Harry Potter and the Philosopher's Stone	Harry Potter and the sorcerer's stone
ISBN	0-7475-3269-9	0747574472 (pbk)	9781338299144 (paperback), 133829914X (paperback)
Subject		Juvenile fiction, fantasy fiction	Wizards—Fiction, Magic-–Fiction, Schools-Fiction
Dewey		823.914	823/.914 [Fic]
Publishing status	• Publication year: 1997 • Publisher: Bloomsbury (UK)	• Publication details: London : Bloomsbury, 1997 2004 printing.	• Published/Produced: New York, NY : Scholastic Inc., [2018]

Data integration challenges

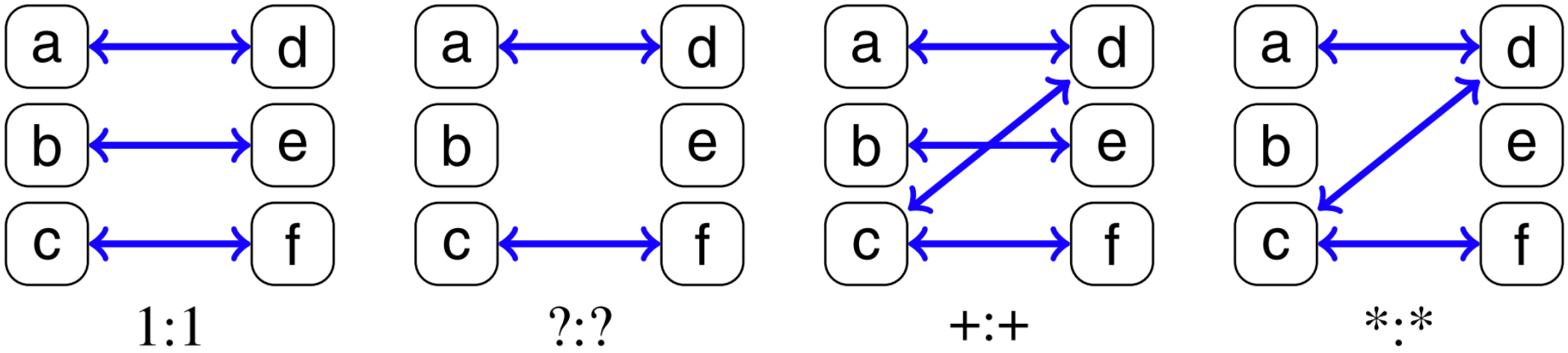
- Structural heterogeneity



Castano S., Ferrara A., Montanelli S., Varese G. (2011) Ontology and Instance Matching. https://doi.org/10.1007/978-3-642-20795-2_7

Data integration challenges

- Multiplicity
  - 1:1, 1:n, m:1, or m:n?
  - This applies to both entities and attributes



Euzenat Jérôme, & Shvaiko, P. (2013). Ontology matching (2nd ed.). Springer Berlin Heidelberg.

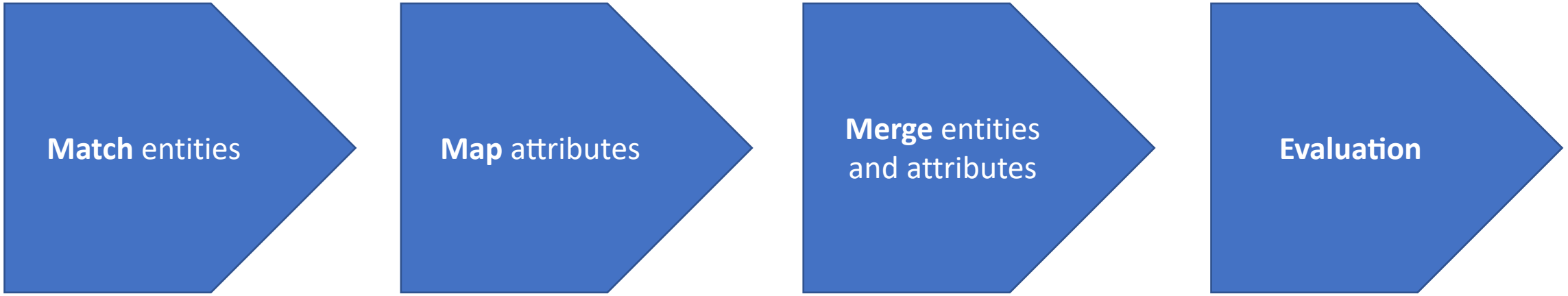
Data integration challenges

- Matching score and threshold
  - <wiki:title> <bl:title> 1.0
  - <bl:Publication_details> <loc:Published_Produced> 0.7
  - False positive vs false negative

- Data sets are too big -> Scalability issues
  - Library of Congress has 19M+ book records, while British Library has 25M+ and WorldCat 400M+!
  - Streaming-based approach might be needed
  - Optimizations
    - Reduction of the number of comparisons
    - Reduction of the cost of each comparison

Goal of Data Integration (DINT)

- Develop a global data integration pipeline

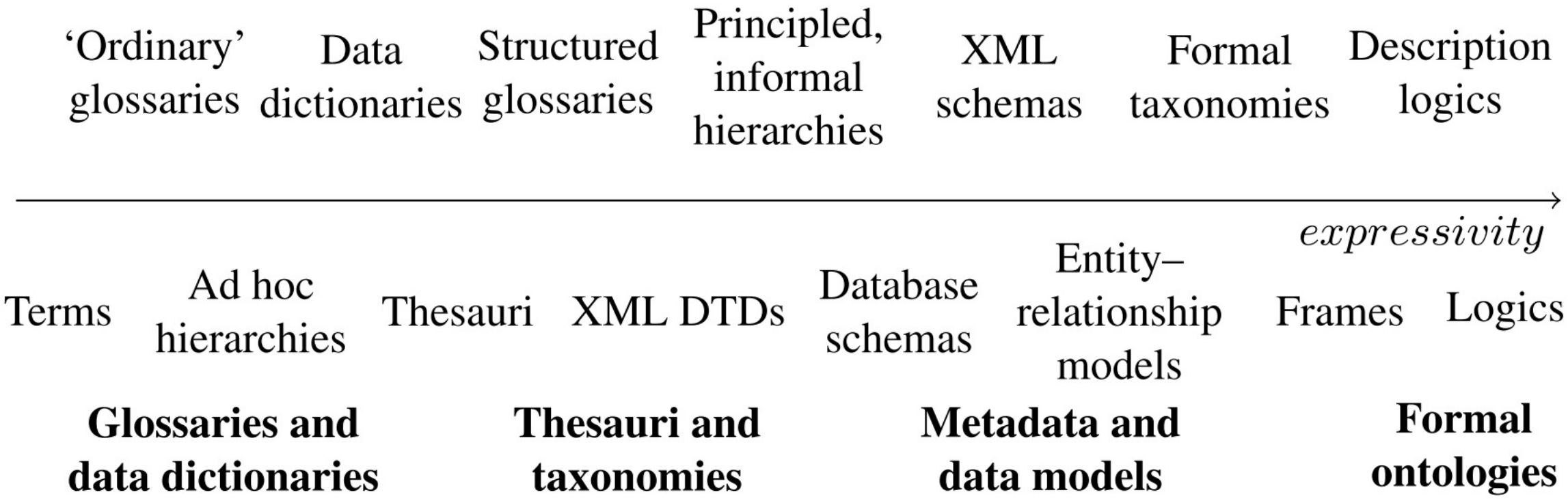


Ontology matching

What is an ontology?

- An ontology is a formal description of knowledge as a set of concepts within a domain and the relationships that hold between them.
- An ontology typically provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary
- Depending on the precision of this specification, the notion of ontology encompasses several data and conceptual models, including, sets of terms, classifications, thesauri, database schemas, or fully axiomatized theories.

Domain modelled with different expressivity

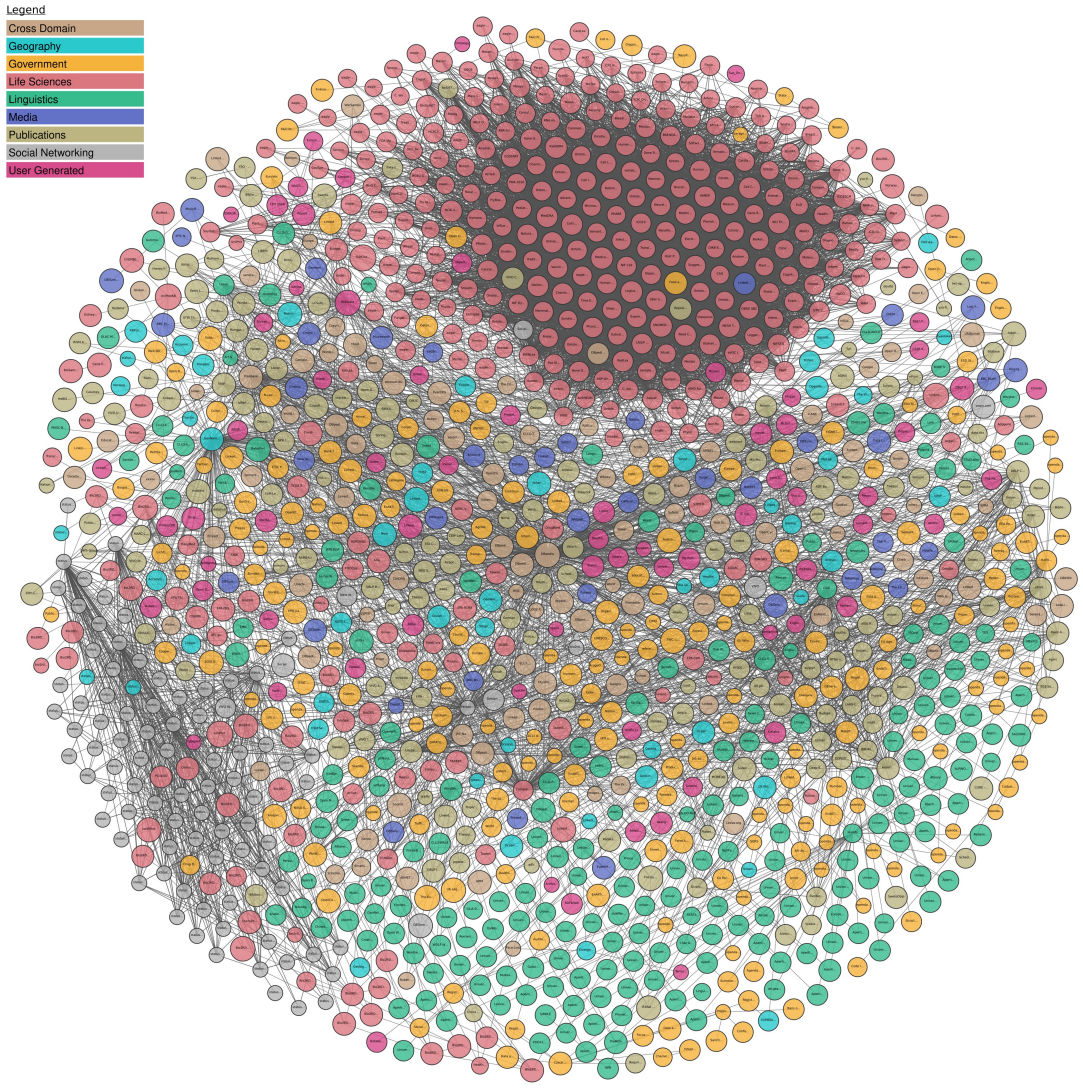


Euzenat Jérôme, & Shvaiko, P. (2013). Ontology matching (2nd ed.). Springer Berlin Heidelberg.

Ontologies are not Reality

- Ontologies are a context-dependent projection (model) of the Reality
- Different ontologies might model the same (similar) or highly related domains, but they might
  - Reflect different tasks and requirements of applications
  - Follow different conventions and restrictions

Linked Open Data cloud



The Linked Open Data Cloud from lod-cloud.net

So it is a mess, but a meaningful one!

Legend

- Cross Domain
- Geography
- Government
- Life Sciences
- Linguistics
- Media
- Publications
- Social Networking
- User Generated

https://lod-cloud.net/

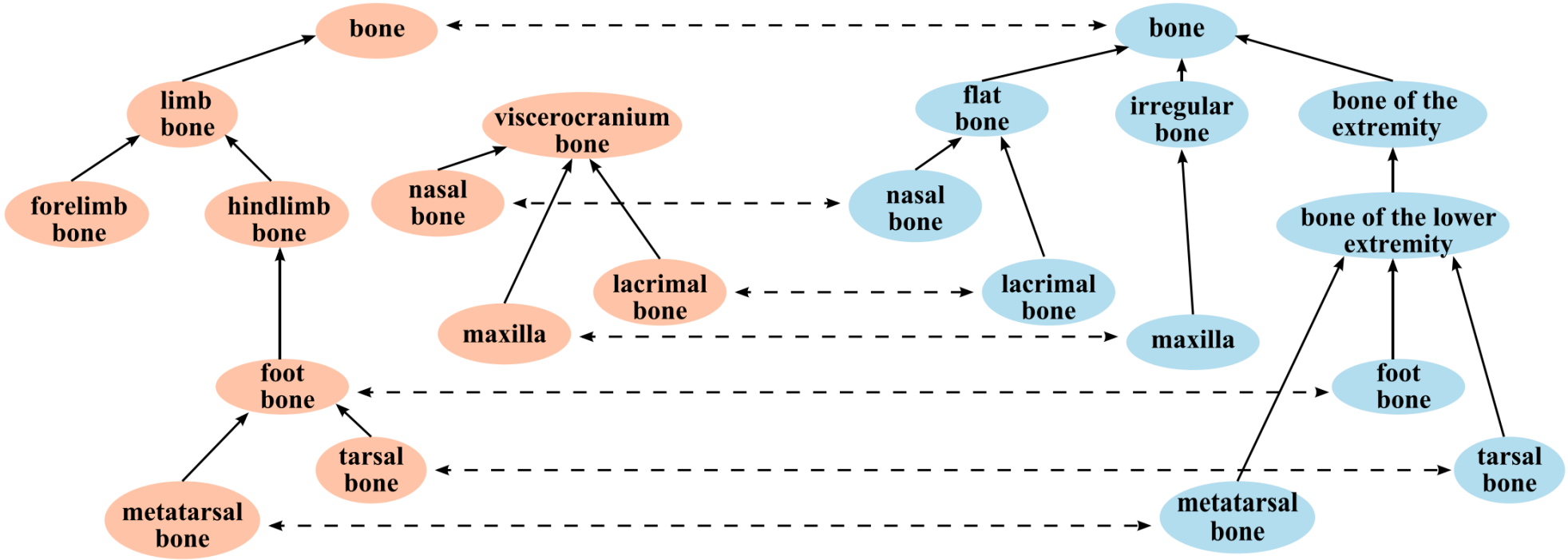
The heterogeneity problem

- Syntactic heterogeneity
  - Using different ontology languages (e.g. XML, OWL)
- Terminological heterogeneity
  - Different terms refer to the same concept
  - Same term describes different concepts
- Conceptual (Semantic) heterogeneity
  - Difference in *granularity, coverage, perspective*
- Semiotic (Pragmatic) heterogeneity
  - Different interpretations wrt different context

Ontology matching to reduce **terminological** & **conceptual** heterogeneity

What is ontology matching?

- Ontology matching, is the process of determining correspondences between entities in different ontologies.



Pambrix, Patrick & Ivanova, Valentina. (2013). A unified approach for debugging is-a structure and mappings in networked taxonomies. Journal of biomedical semantics. 4. 10. 10.1186/2041-1480-4-10.

Correspondence

- Given two ontologies o and o', a correspondence between o and o' is defined as $< id, e, e', r, n >$,

  where

  - id is a unique identifier of the correspondence
  - e and e' are entities of o and o' respectively, e.g., classes, instances
  - r is a relation, e.g., equivalence($=$), more general ($\sqsupseteq$), more specific ($\sqsubseteq$), disjointness ($\perp$), etc.
  - n is a confidence measure (typically in a range of [0,1]) for the correspondence between e and e'.

Ontology matching

- A schema is a structure of metadata describing how data, i.e., instances, can be stored, accessed, and interpreted by users and applications.
- Schema matching
  - Whether two concepts (e.g., *book* vs *manuscript*) or two properties (e.g. *birthplace* vs *hometown*) are the same
- Instance matching
  - Whether different instances refer to the same real-world entity in a given domain (e.g., a person, a place, a movie, a book, etc.)

Ontology matching applications

Application	instances	run time	automatic	correct	complete	operation
Ontology evolution	√			√	√	transformation
Schema integration	√			√	√	merging
Catalog integration	√			√	√	data translation
Data integration	√			√	√	query answering
Linked data	√			√		data interlinking
P2P information sharing		√				query answering
Web service composition		√	√	√		data mediation
Multi agent communication		√	√	√	√	data translation
Query answering	√	√				query reformulation

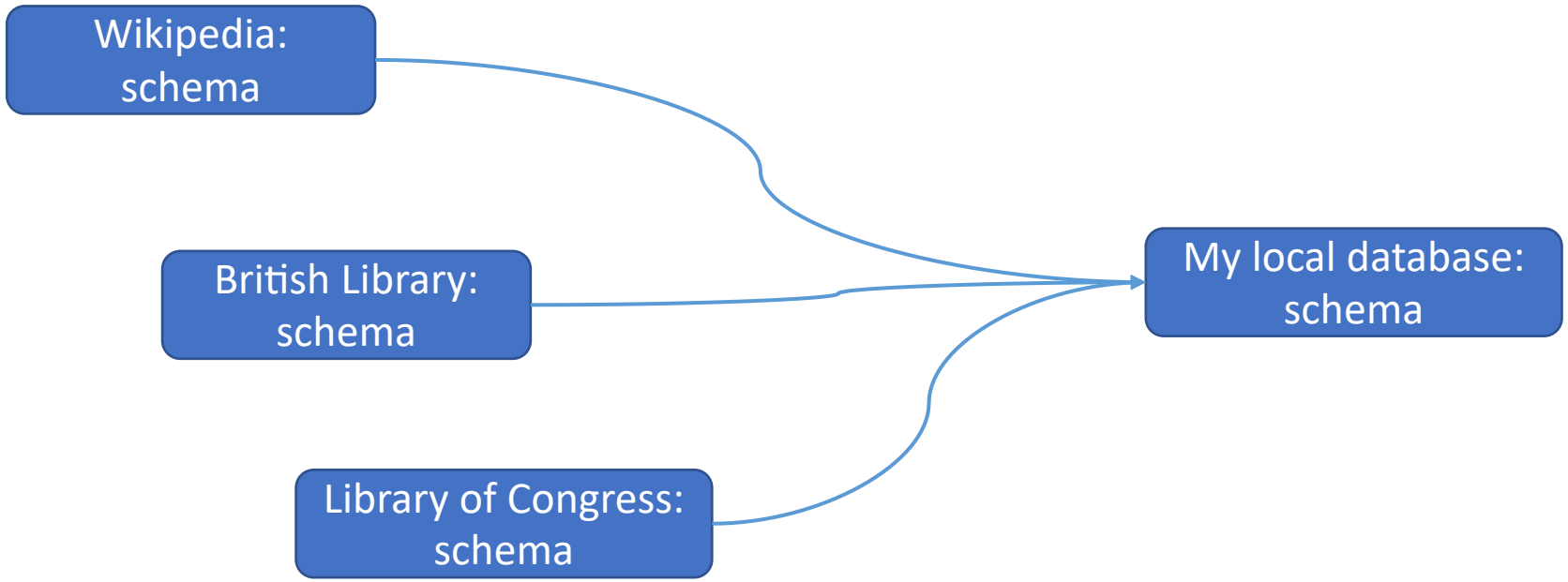
Data Integration

- Schema matching focuses on finding the correspondence among schema elements in two semantically correlated schemata
- Schema mapping describes how a source database schema relates to a target database schema
- Record linkage (also known as entity resolution and deduplication) identifies records that refer to the same logical entity
- Data fusion focuses on resolving conflicts and determining the true data values, leveraging information in heterogeneous data sources

X. Wang, L.M. Haas and A. Meliou, Explaining Data Integration, IEEE Data Eng. Bull. 41(2) (2018), 47–58

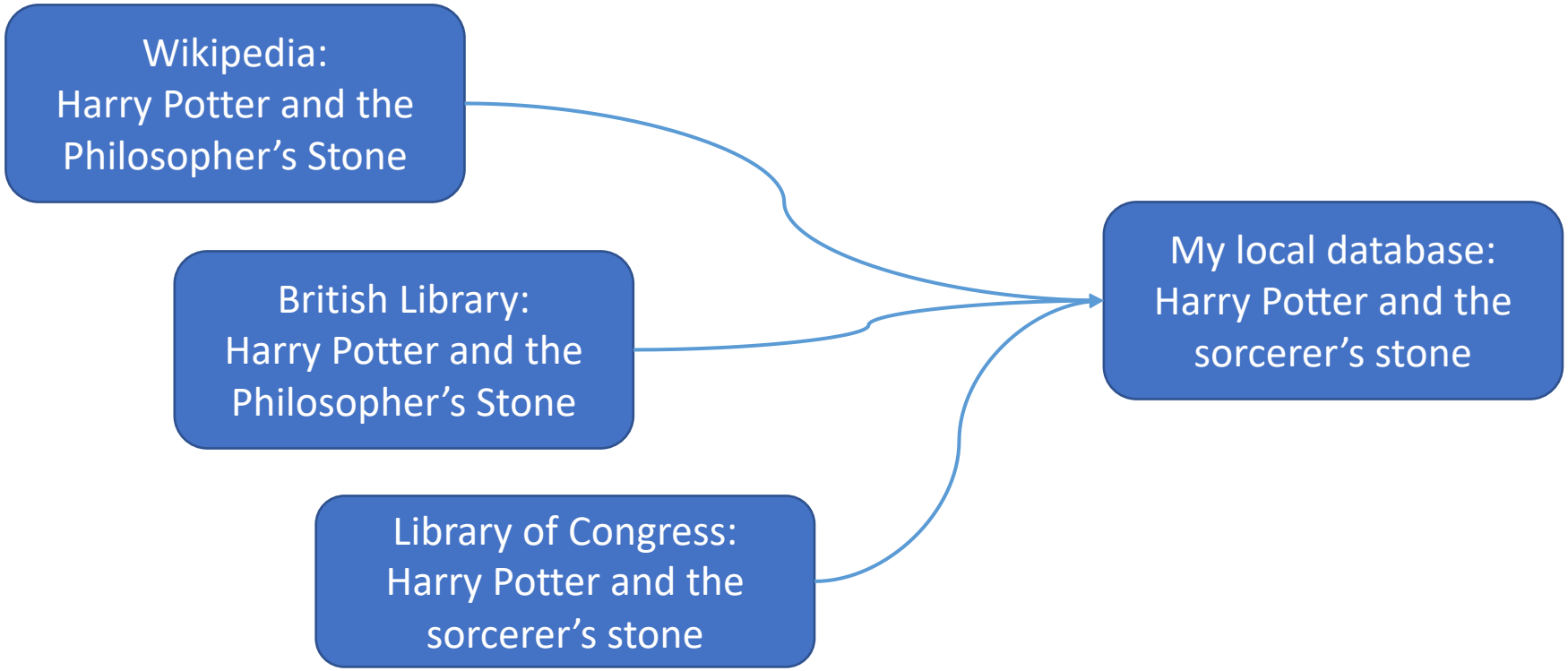
Schema integration

- Schema integration requires the ability to merge schemas under consideration into a single schema

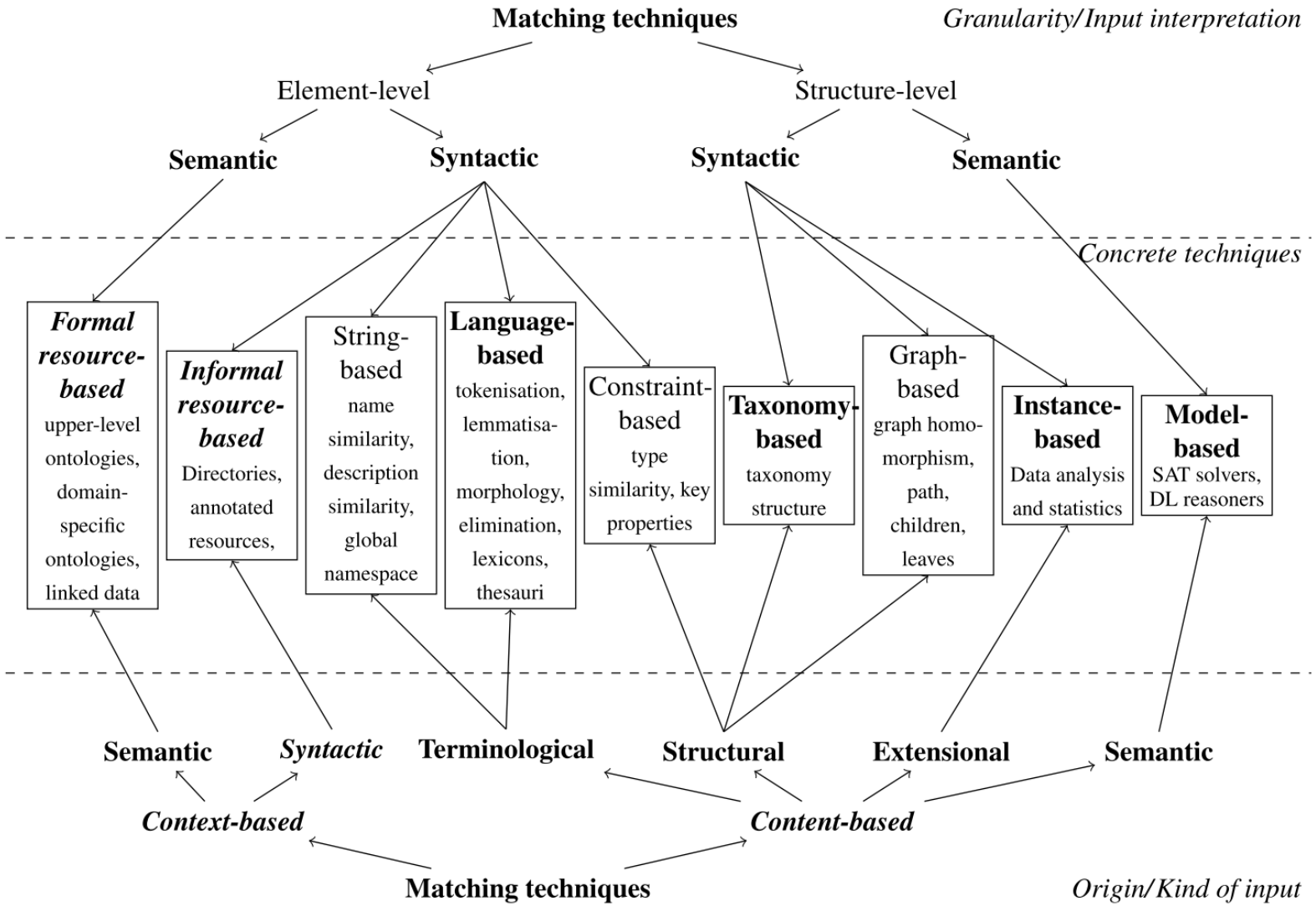


Instance integration

- Instance (data) integration requires the ability to translate data instances residing in multiple local schemas according to a global schema definition



Classification of schema matching methods



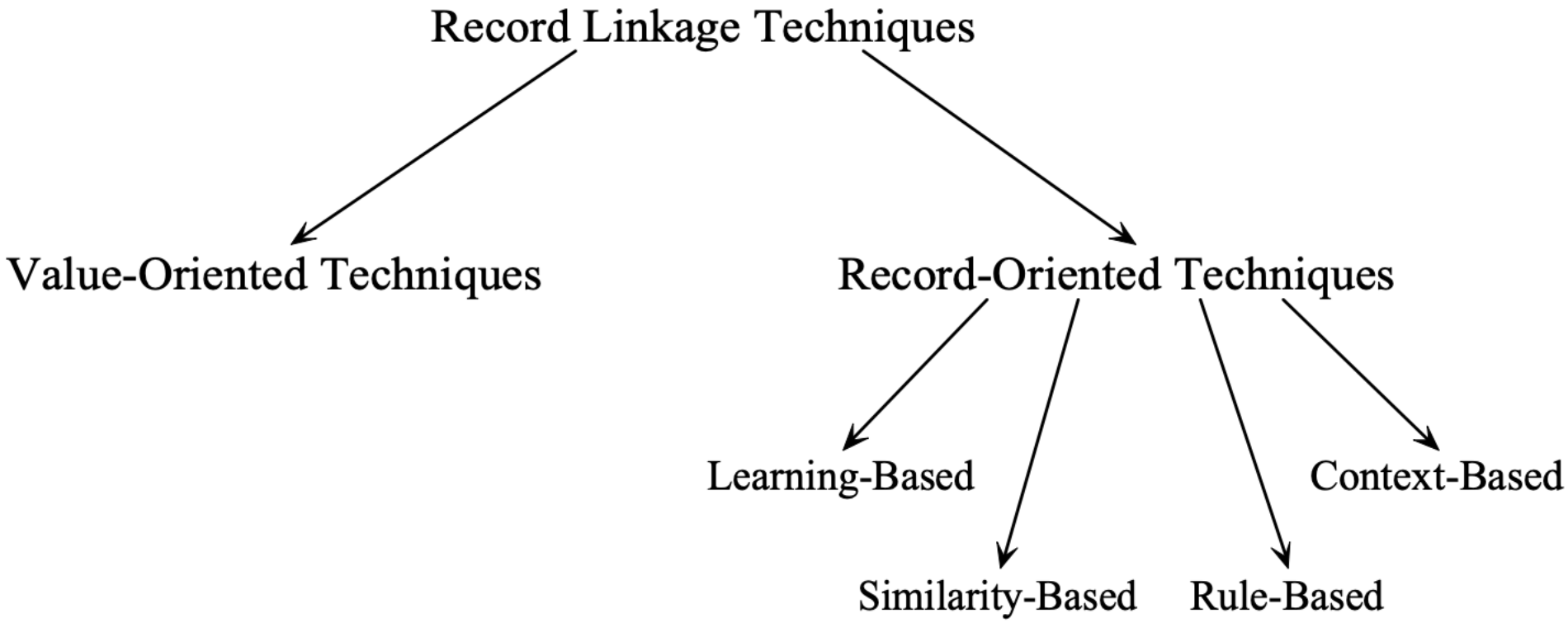
Euzenat Jérôme, & Shvaiko, P. (2013). Ontology matching (2nd ed.). Springer Berlin Heidelberg.

Categories of matching techniques

- Element-level techniques consider ontology entities or their instances in isolation from their relations with other entities or their instances
  - String-based, linguistic-based, phonetic-based, etc.
- Structure-level techniques consider the ontology entities or their instances to compare their relations with other entities or their instances.
  - Graph-based, taxonomy-based, instance-based, etc.

Instance matching and integration

Instance matching (record linkage)



Castano S., Ferrara A., Montanelli S., Varese G. (2011) Ontology and Instance Matching. https://doi.org/10.1007/978-3-642-20795-2_7

Value-oriented techniques

- Assumption: the similarity level of two records (entities) can be derived by matching the values of their comparable attributes

- Mostly focus on similarity of string attributes
  - Character-based: Edit Distance, Smith-Waterman Distance, Jaro Distance
    - Typographical variations, e.g. “organisation” vs “organization”
  - Token-based: Bag-of-Word
    - Different conventions for describing data, e.g. “J.K. Rowling” vs “Rowling, J. K.”
  - Linguistic-based: use NLP, lexicons, or domain specific thesauri to match words based on linguistic relations (homonymy, synonymy, partonomy, etc) or exploiting morphological properties
  - Phonetic-based: Soundex, NYSIIS, Metaphone
    - Phonetic similarity, e.g. “Kageonne” vs “Cajun”

String comparison: character-based

- Levenshtein distance represents the number of insertions, deletions, and substitutions required to change one word to another.
  - For example, the Levenshtein distance between "kitten" and "sitting" is 3:
    - **k**itten → **s**itten (substitution of "s" for "k"),
    - sitt**e**n → sitt**i**n (substitution of "i" for "e"),
    - sittin → sittin**g** (insertion of "g" at the end).

- Damerau-Levenshtein distance counts transpositions as a single edit
  - For example, damerau_levenshtein_distance('fish', 'ifsh') == 1 while levenshtein_distance('fish', 'ifsh') == 2

String comparison: character-based

- Hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are different.
  - For example, the Hamming distance between "karoln" and "kerstn" is 3.

- N-gram
  - Takes as input two strings and calculate the number of the common n-grams between them, normalised by max(length(string1), length(string2))
  - Bigrams for matching is ma, at, tc, ch, hi, in, ng
  - Bigrams for mapping is ma, ap, pp, pi, in, ng
  - Similarity('matching', 'mapping')=3/8

String comparison: linguistic-based

- Tokenization
  - Parses names into tokens by recognizing punctuation, cases
  - string-based methods -> [string, based, methods]
- Lemmatisation
  - Analyses morphologically tokens to find their basic forms
  - methods -> method
- Remove stop words
  - a, he, them, by, from

Phonetic encoding

- American soundex
  - Soundex is a phonetic algorithm for indexing names by sound, as pronounced in English.
    - soundex(‘Ashcraft') == soundex(‘Ashcroft') == ‘A261’
    - Soundex('Rupert') == soundex('Robert') == 'R163’
  - The goal is for homophones to be encoded to the same representation so that they can be matched despite minor differences in spelling.

Classification of matching vs non-matching

- When the similarity of each pair of corresponding attribute values is computed, a decision engine is needed to classify whether two entities match or not
  - Learning-based
  - Similarity-based
  - Rule-based
  - Context-based

Learning-based techniques

- Supervised learning
  - Training data: a set of instance pairs and their expected classification (i.e. matching or non-matching records)
  - Non-trivial to get a high-quality and balanced training data set
  - Manually adding ambiguous cases if possible
- Unsupervised learning
  - Clustering record pairs with similar features that belong to the same class (i.e. matching or non-matching records)
- Combining different learning techniques

Similarity-based techniques

- Considering the input records as long attribute values
  - Concatenate all attribute values into one single string
- Average similarity of each pair of corresponding attribute values
  - Weighted average may better reflect domain knowledge
- Taking into account the frequency each value occurs
  - A matched “Paul Smith” counts less than the matched “Bamidele Melisizwe”

Rule-based techniques

- Instead of similarity values, a Boolean output is assigned
  - For example, if two records denoting books share the same value on attributes “Title”, “Author” and “Publication year”, there is a very high probability that the considered records refer to the same book.

Context-based techniques

- Taking into account the relationships between records
  - For example, matching clusters of records instead of individual records

Optimisation

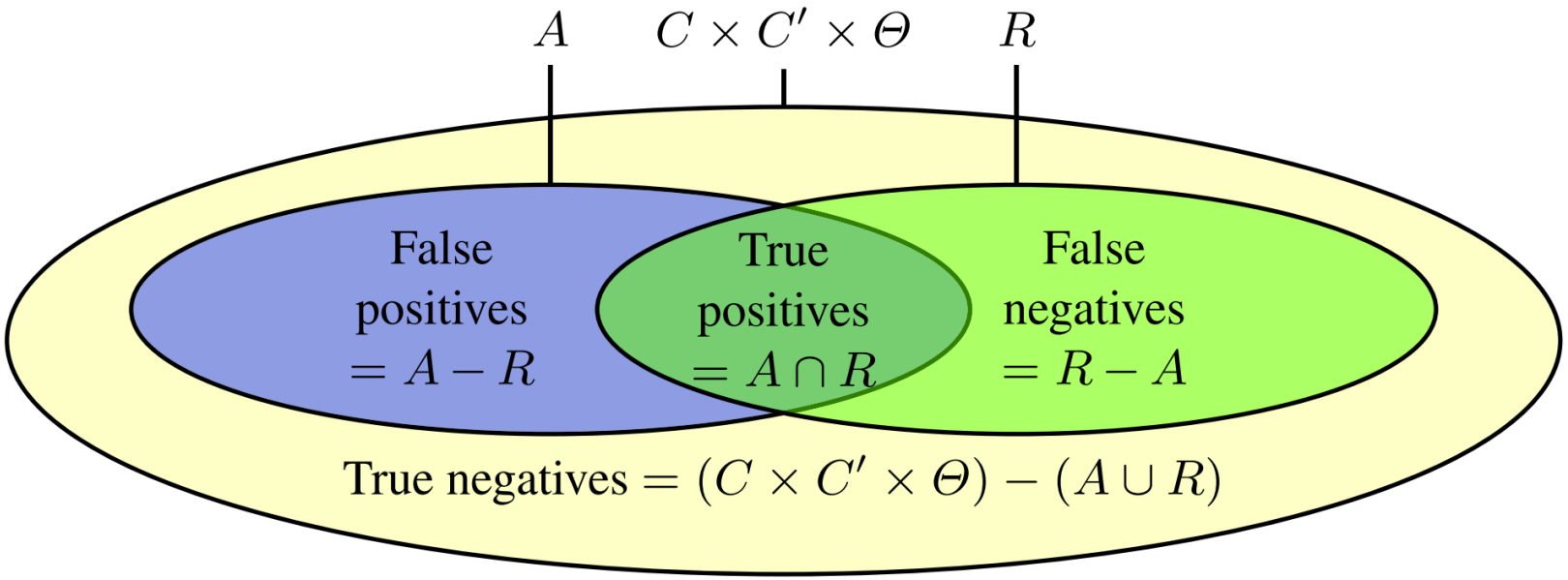
- Reduction of the number of comparisons
  - Blocking: dividing instances in homogeneous and mutually exclusive *blocks* and comparing only with instances belonging to the same block
  - Sorted neighbourhood: sorting instances according to the value they assume on the property with the highest discriminating power and only comparing instances within a shifting- window of a fixed dimension
- Reduction of the cost of each comparison
  - Only a subset of the corresponding attribute values matters
    - E.g. title-author-year is a reasonable subset to distinguish books

Similarity filter and alignment extraction

- Many algorithms are based on similarity or distance computation. A few operations can be based on similarity/distance matrices.
- Various thresholding options to filter similarities
  - Hard threshold retains all the correspondence above threshold n;
  - Delta threshold consists of using as a threshold the highest similarity value out of which a particular constant value d is subtracted;
  - Gap threshold retains the correspondences ordered by decreasing similarity until the difference in similarity between two correspondences becomes larger than n;
  - Proportional threshold consists of using as a threshold the percentage of the highest similarity value;
  - Percentage retains the n% correspondences above the others.

Evaluation

Evaluation with a reference alignment



- Precision measures the ratio of correctly found matches (true positives) over the total number of returned matches (true positives and false positives): $\frac{A \cap R}{A}$
- Recall measures the ratio of correctly found matches (true positives) over the total number of expected matches (true positives and false negatives): $\frac{A \cap R}{R}$
- F-measure = 2 * P * R / (P + R)

Euzenat Jérôme, & Shvaiko, P. (2013). Ontology matching (2nd ed.). Springer Berlin Heidelberg.

Evaluation without a reference alignment

- Sampling or pooling-based evaluation
  - Build a subset of alignment using sampling or pooling
  - Expert evaluation
- End-to-end evaluation
  - Testing the quality of alignment using an application
- Performance measures
  - Speed, memory, scalability

DINT assignment

Data sets

- Tvguide-ds.xml
  - A data set with information about 100 top movies from a TV guide. The data set contains limited general information about these movies (name, year, rating, running-time, country, genres, actors) as well as TV-guide specific information (rank, format, user-rating, reviews).

- Imdbfull-ds.xml
  - An excerpt of about 250.000 movies from the well-known IMDB knowledge base. The data set has much more general information about movies.

Your task

Develop a data integration pipeline enriching the information of the TV guide movies with additional information available in the IMDB data set.

- For each movie in the TV guide, find a best match from IMDB, based on the string similarity of the titles.
- Evaluate your matches
- Mapping attributes and apply value-oriented techniques to find matches. Evaluate your matches
- Merging corresponding attribute values

Practical tips

Try a few different similarity measures over attribute values and see how they influence the final precision and recall.

- Use Python package jellyfish for string similarity functions.

String comparison:

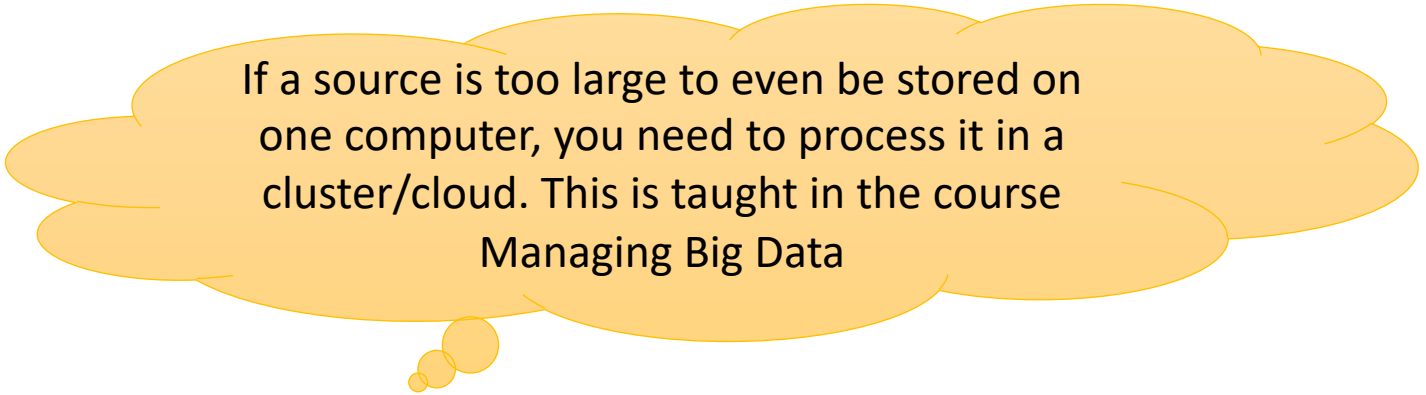
- Levenshtein Distance
- Damerau-Levenshtein Distance
- Jaro Distance
- Jaro-Winkler Distance
- Match Rating Approach Comparison
- Hamming Distance

Phonetic encoding:

- American Soundex
- Metaphone
- NYSIIS (New York State Identification and Intelligence System)
- Match Rating Codex

DOM vs SAX

- DOM stands for Document Object Model. The DOM API provides the classes to read and write an XML file. DOM reads an entire document.
- SAX represents Simple API for XML, which is suitable for large XML files because it doesn't require loading the whole XML file.
- Therefore,
  - Read the whole TV guide source into main memory using the XML DOM method.
  - Read the IMDB source one movie at-a-time using the XML SAX method.



Processing XML data with DOM

- For inexperience programmers: `xml.etree.ElementTree`

```
import xml.etree.ElementTree as ET

tvguide = ET.parse('tvguide-ds.xml').getroot() # parse the doc and get
its root
print (”#movies = ", len(tvguide)) # total number of the movie
for movie in tvguide: # loop all movies
   ...
```

- For more experience programmers: `xml.dom.minidom`

```
import xml.dom.minidom

dom = parse('tvguide-ds.xml’).firstChild # parse the doc and get its root
print (”#movies = ", dom.getElementsByTagName("movie").length) # total
number of the movie
for movie in dom.childNodes: # loop all movies
   ...
```

Streamed processing of large data with SAX

```
from xml.sax import parse
from xml.sax.handler import
  ContentHandler

class imdb_Handler(ContentHandler):
  def __init__(self, total_movies):
    self.movie_key = 1
    self.total_movies = total_movies

  def startElement(self,name,attrs):
    if name == "movie":
      ...

  def endElement(self, name):
    if name == "movie" and
self.movie_key>=self.total_movies:
      raise Exception("")
    elif name == "movie" and
```

```
self.movie_key<self.total_movies:
      self.movie_key+=1
    else:
      pass

  def characters(self, content):
    if content.strip() != "":
      ...


try:
  parse("imdb-ds.xml", imdb_Handler(5))
except:
  pass
```

Observe that we do not store the document in main memory; we only do some book keeping.

About merging

- For easy cases:
  - Keep attributes and their values in IMDB that are not in the TV guide
- For single-valued attributes (e.g., title, year)
  - If there is an inconsistency, take the more authoritative one (in our case, the attribute values in IMDB are probably more reliable)
- For multi-valued attributes (e.g., genre)
  - Take the union as any of these values is potential true
- For attributes that may contain ambiguous values (e.g., actor names)
  - Take the union but be careful not to introduce duplicates (e.g., variations of person names)

Highly recommended in practice

- Always develop alternative approaches and evaluate them properly
- Use systematic approach to set parameters (e.g., weights, thresholds)
- Be aware, any hand-made *ground-truth* is prone to human biases.

Good luck and have fun!