UNIVERSITY
OF TRENTO - Italy

Know Dive.

Towards the integration of KOS with the Linked Data Cloud

Vincenzo Maltese, Feroz Farazi
DISI – University of Trento

The main message

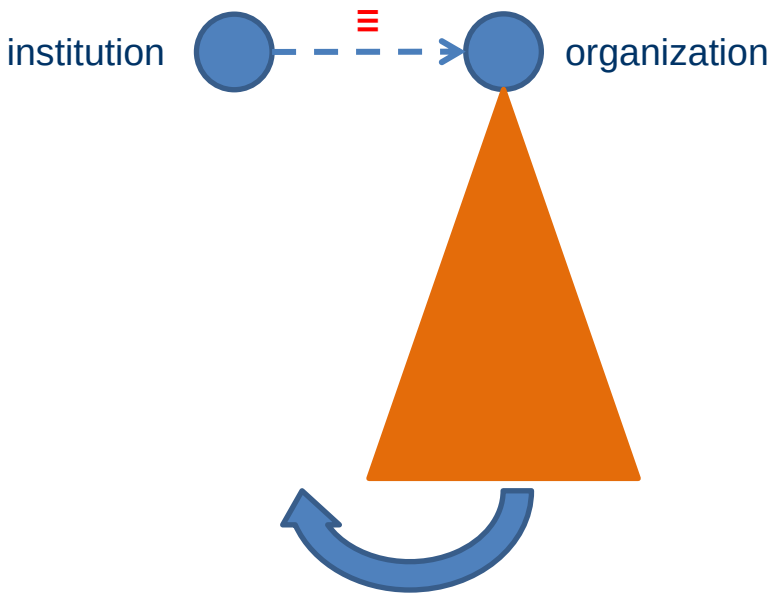
Reusing and sharing ontologies: the linked data initiative

In reusing and sharing ontologies:

1. **It is fundamental to take into account and make explicit their purpose and semantics**
2. **The difference in the purpose is reflected in the difference in the semantics and in the applications we enable**
3. **Before integrating them, it is possible and essential to translate them such that they have the same semantics**
4. **To maximize reuse, publishing an ontology requires appropriate semantics**

Building a KOS can be extremely costly

- **Reuse as much as possible**
  - **Need**: discover similar resources and (partially) integrate them
  - **Solution**:
    - Manual approaches (accurate but slow)
    - Automatic tools (need background knowledge, need manual validation, but fast)



- **Share as much as possible**
  - **Need**: incentives, a common framework
  - **Solution**: the Linked Data initiative

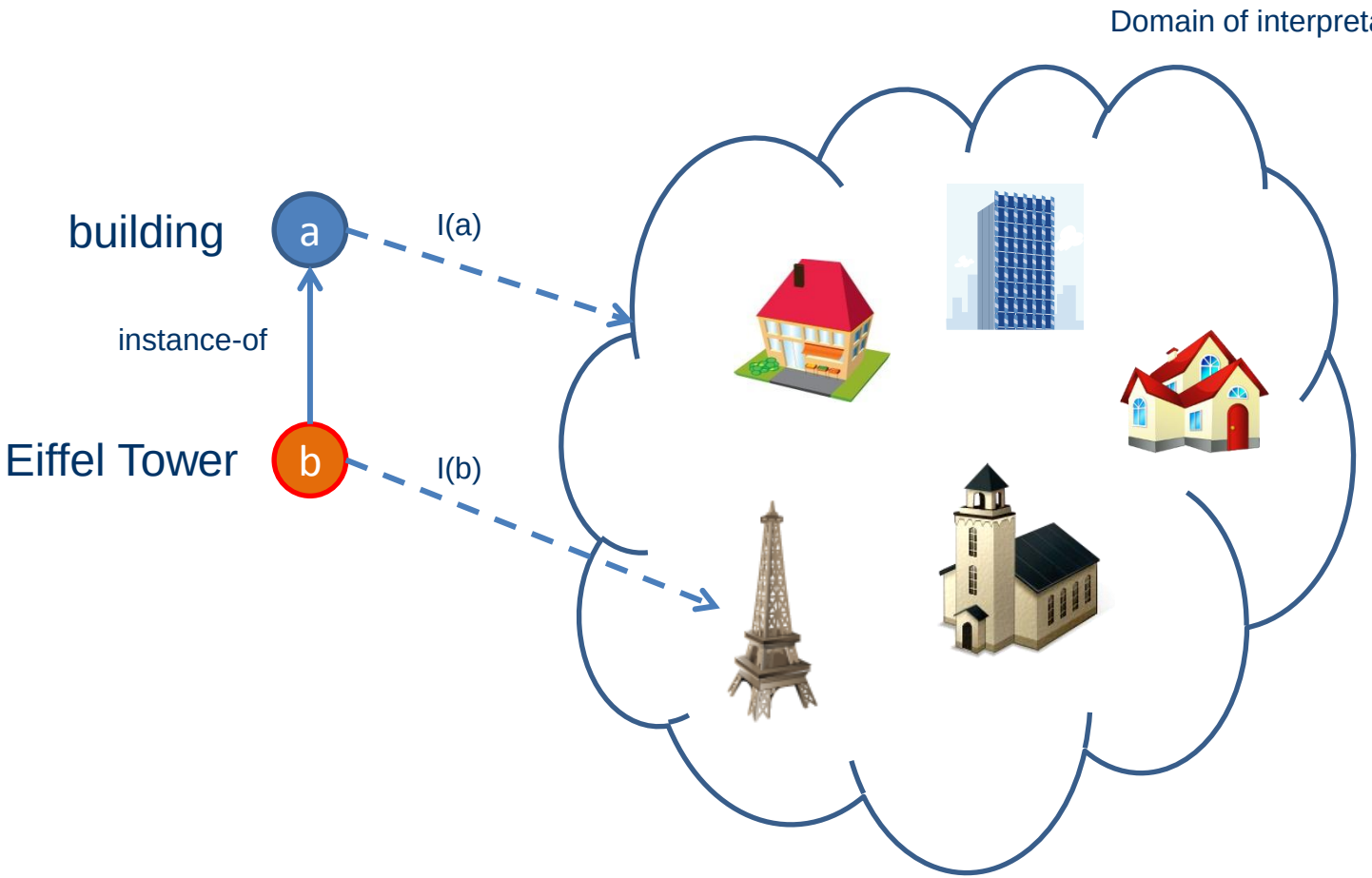
Ontologies are extremely diverse

They may differ in scope, purpose, structure, terminology, language, coverage, formality and conceptualization

In sharing and reusing them, it is fundamental to take into account and make explicit:

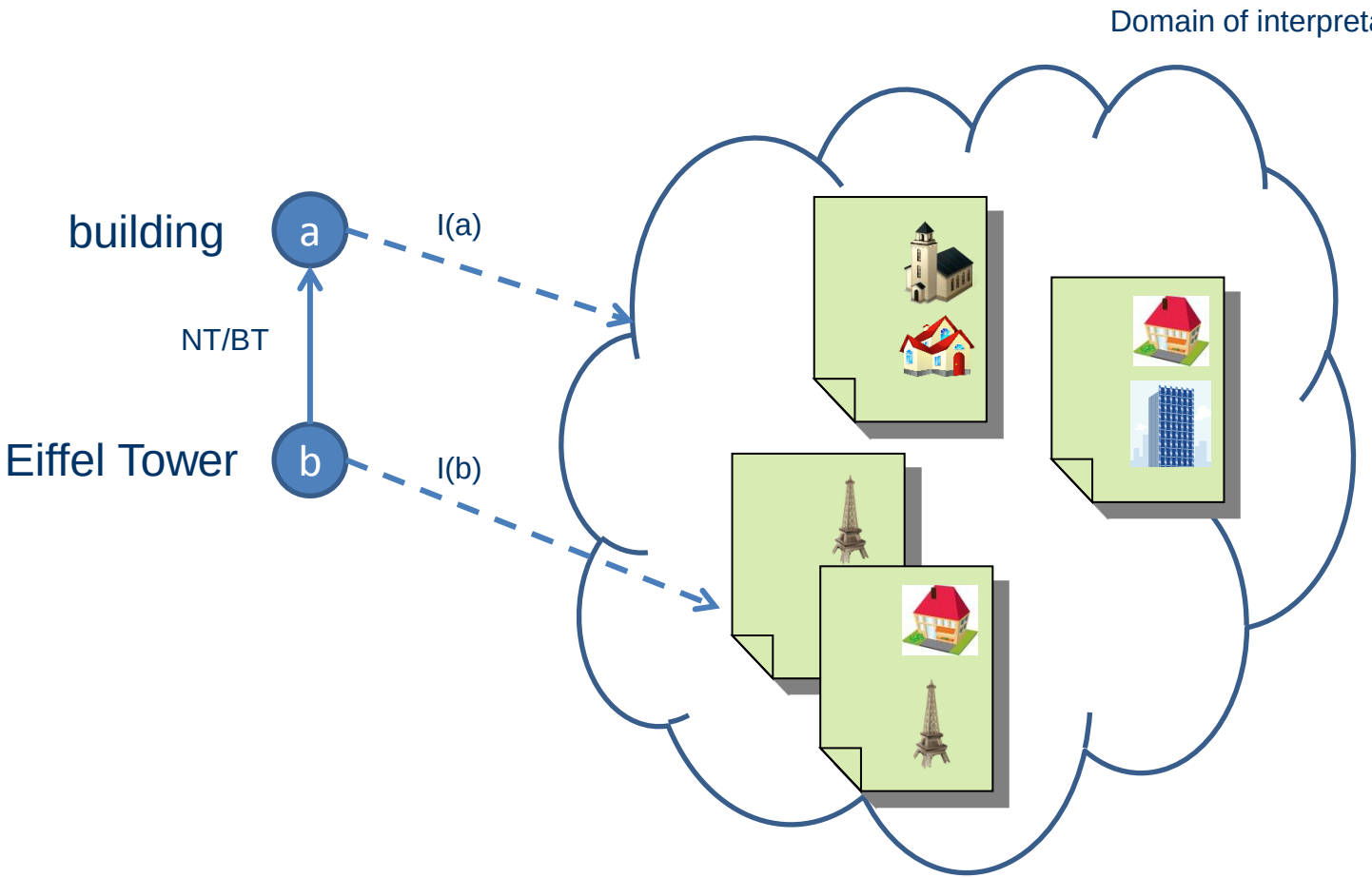
- **The difference in the purpose (their goal)**
- **The difference in the semantics (their meaning)**

Real World semantics



Each term at nodes denotes a real world object or a set of objects

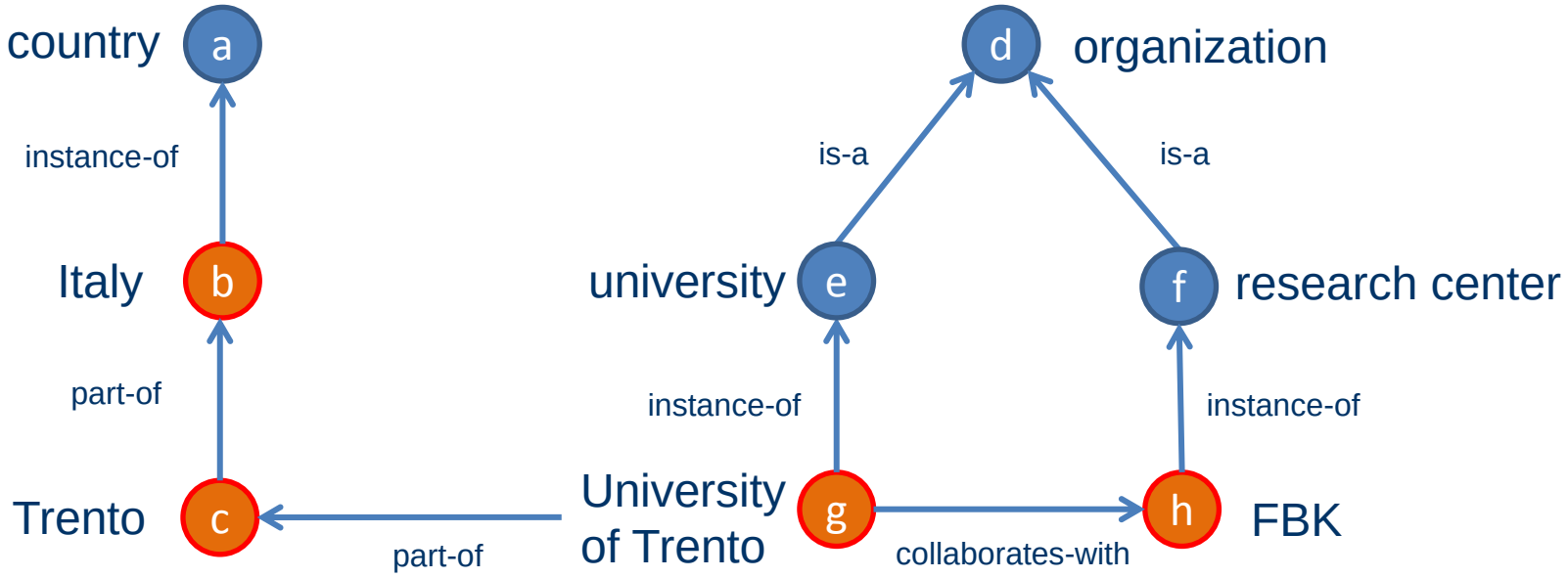
Classification semantics



Each term at nodes denotes a set of documents about real world objects

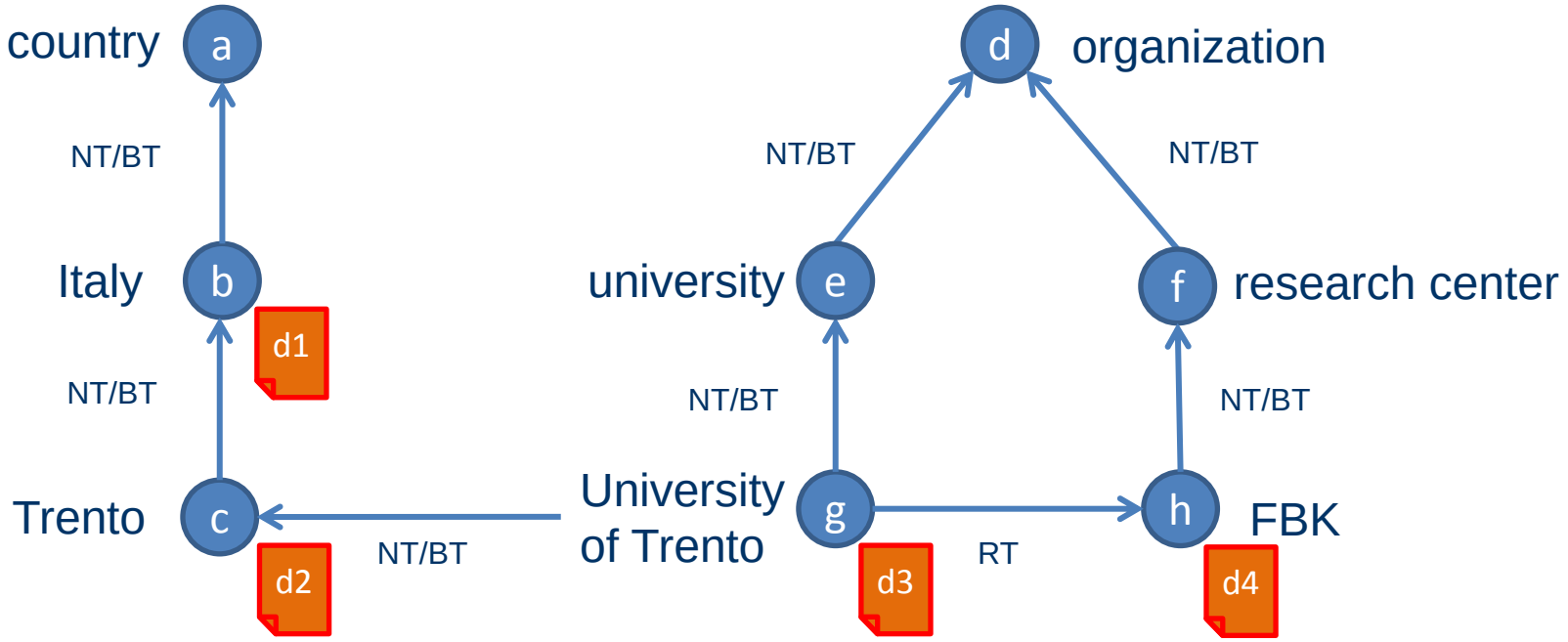
Difference in the purpose is reflected in different semantics

Descriptive ontologies



- **Purpose**: describing a domain
- **Semantics**: real world semantics

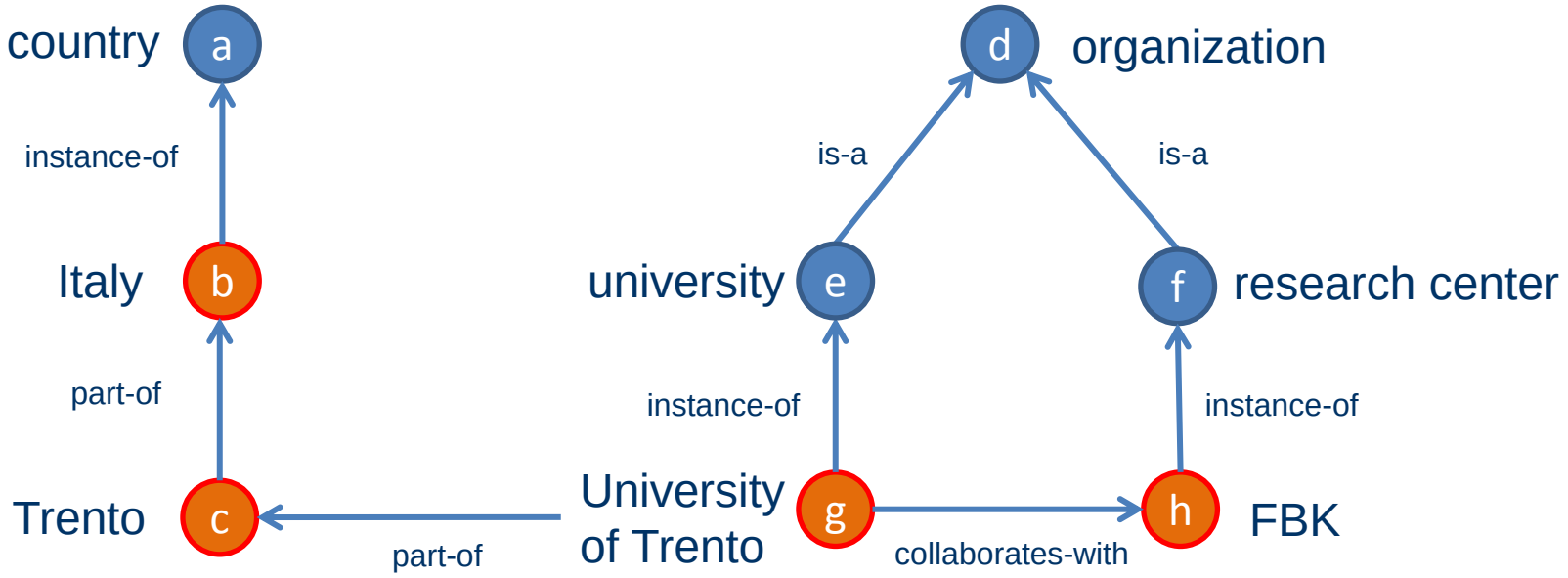
Classification ontologies



- **Purpose**: classifying, searching and browsing documents
- **Semantics**: classification semantics

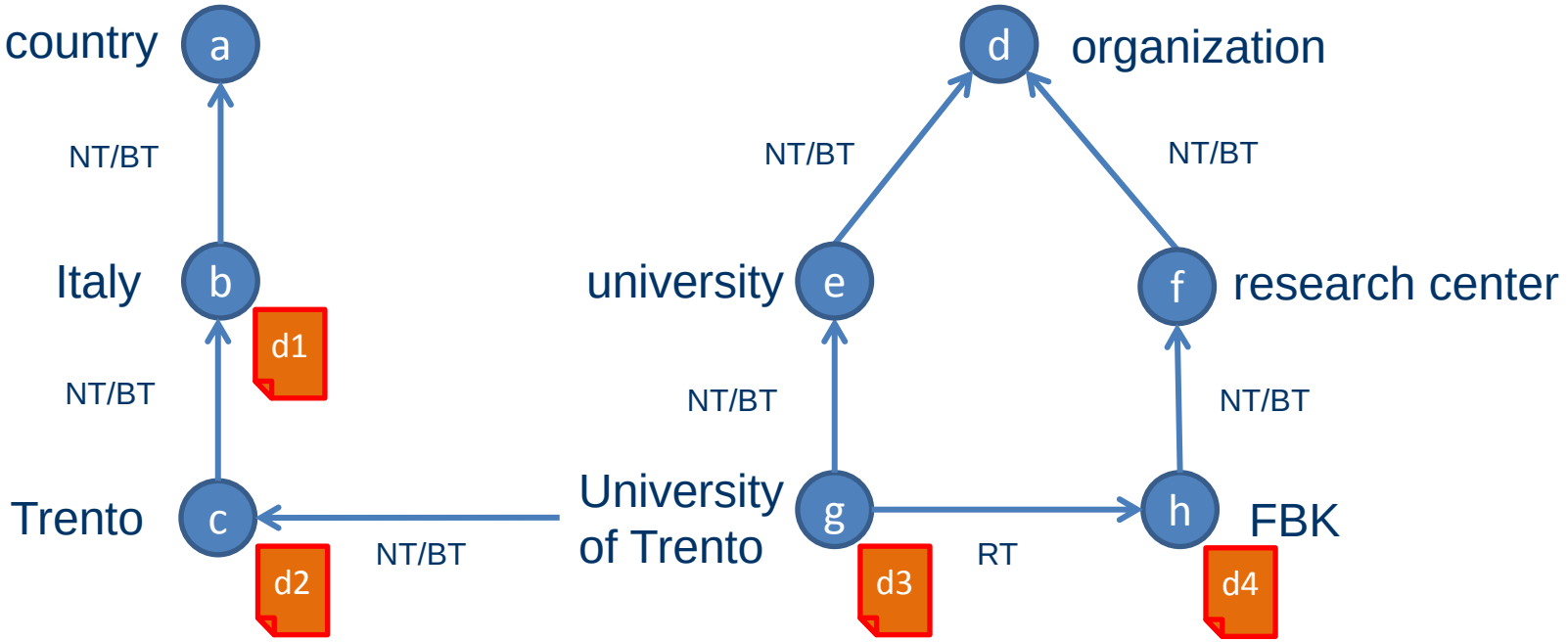
Difference in the semantics affects what we can do with them

Descriptive ontologies: typical queries



- **Give me all the countries:** {Italy}
- **Give me all the organizations:** {University of Trento, FBK}
- **Give me all the organizations located in Italy:** {University of Trento}

Classification ontologies: typical queries

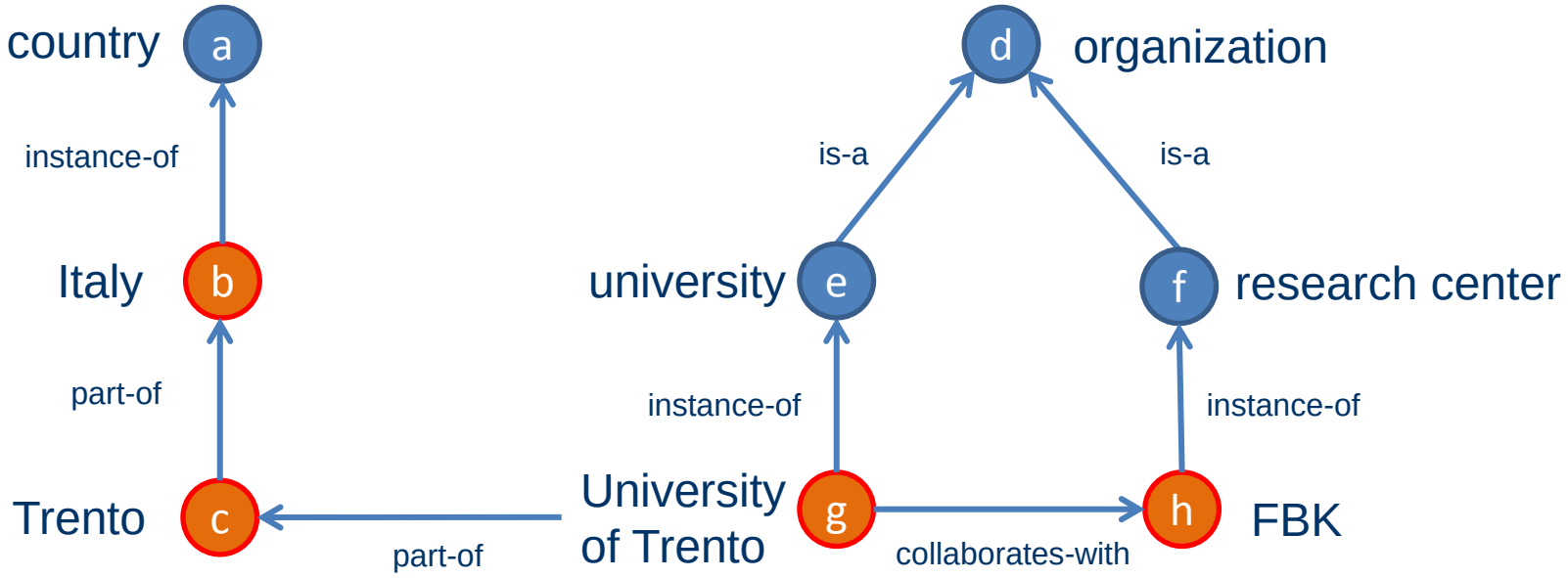


- **Give me all documents about countries**
- **Give me all documents about Italy:** {d1} or {d1, d2} or{d1, d2, d3}?

  It depends! Do we expand? Are all NT/BT transitive?

Making explicit the semantics allows for automation

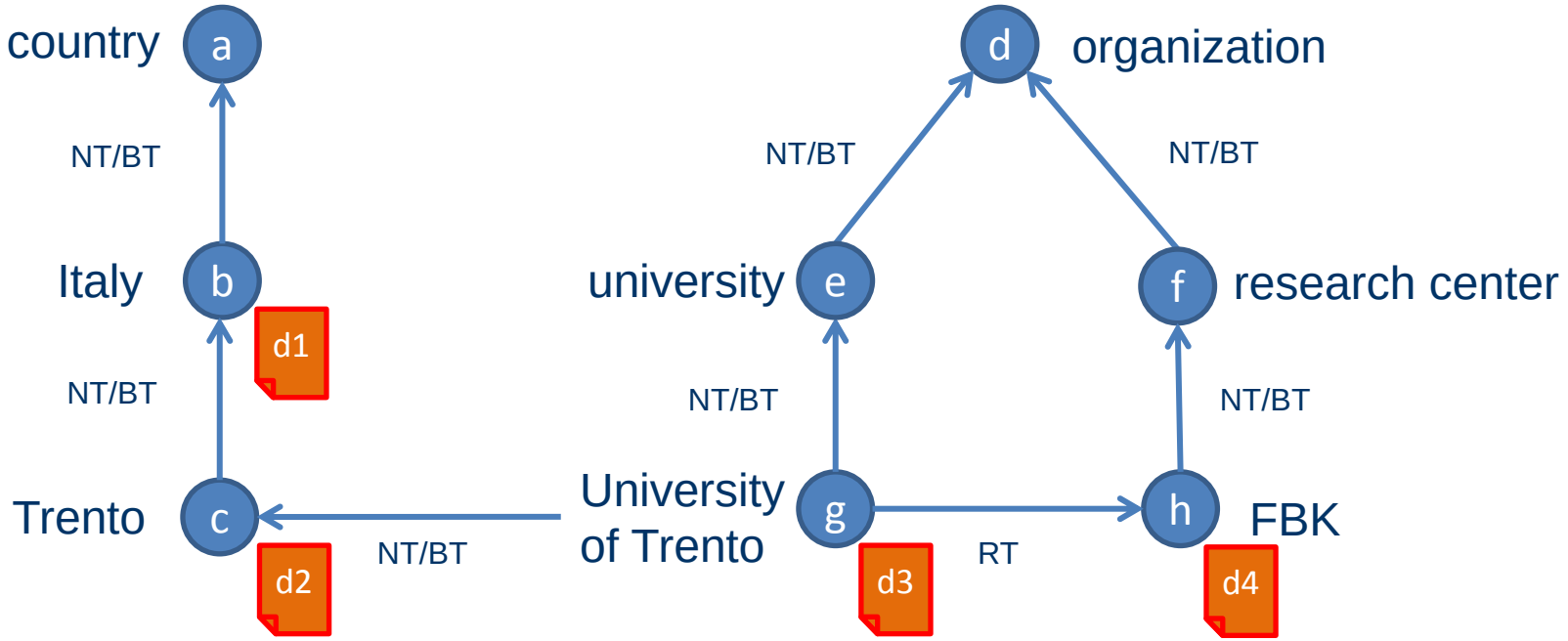
Descriptive ontologies: how to make them formal



Assume we use Description Logics (DL):

- **Classes correspond to concepts**
- **Instances correspond to individuals**
- **Is-a relations are translated into subsumption (⊑)**
- **Other relations correspond to DL roles**

Classification ontologies: how to make them formal



Assume we use Description Logics (DL):

- **Classes correspond to concepts**
- **Documents correspond to individuals**
- **Transitive NT/BT relations are translated into subsumption ($\sqsubseteq$)**
- **RT and non-transitive NT/BT relations correspond to DL roles**

Converting, integrating and reusing ontologies

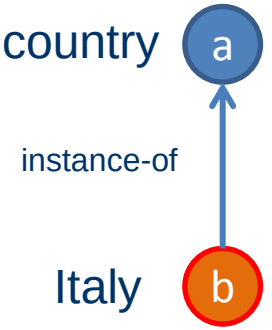
Convert before integrating ontologies

- **It is clearly not appropriate to integrate ontologies having different semantics.**

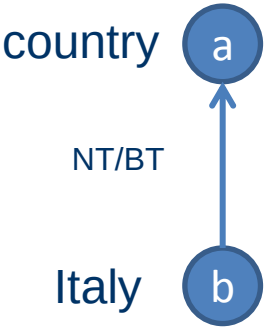
- **Given the purpose select the semantics**
  - **If the purpose is to classify, convert both ontologies into classification ontologies**
  - **If the purpose is to describe, convert both ontologies into descriptive ontologies**

From descriptive to classification ontologies

- **Convert instances to classes**
- **Convert instance-of, is-a and transitive part-of into NT/BT relations**
- **Convert other relations into RT relations**



Hierarchies are constructed on the basis of *genus-species* (is-a, instance-of) and *whole-part* (part-of) relations [Ranganathan, 1967. Prolegomena to library classification]

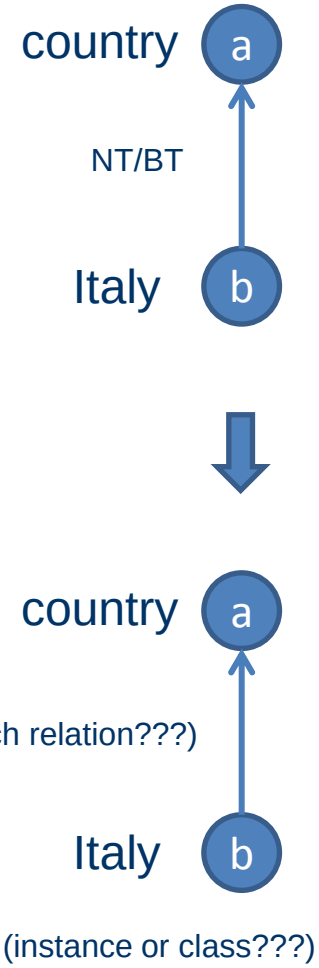


The process above can be easily automated

There is a clear loss of information

From classification to descriptive ontologies

- **Each class has to be mapped to either a real world class or instance**
- **Each transitive NT/BT relation has to be converted into either an instance-of, is-a or part-of**
- **Each RT relation and non-transitive NT/BT relation has to be codified into an appropriate real world associative relation**



The process above CANNOT be automated

A substantial amount of human effort is required

For those reasons

Distributing schemes as descriptive ontologies would ensure maximum reusability

Let us look at a concrete use case…

GeoWordNet: a multilingual descriptive ontology

- **Neat separation between language and conceptual levels**
- **It is currently in English and Italian**
- **Built from WordNet, Italian MultiWordNet and GeoNames**

Objects	Quantity
Classes	110,459
Instances	6,927,078
Instance-of	6,927,078
Is-a	89,266
transitive part-of	5,325
Associative relations	98,907

- **We compute the transitive closure for both the descriptive and classification version of the ontology**
- **We use the appropriate semantics according to the task**

Conclusions

- **There is the need to reuse/share ontologies**
- **It is fundamental to take into account and make explicit their purpose and semantics**
- **Particular attention has to be paid to the transitivity of the relations**

- **Storing ontologies in their descriptive version maximizes reuse and effectiveness**

Towards the integration of KOS with the Linked Data Cloud

Thank you for your time and interest!

Questions?

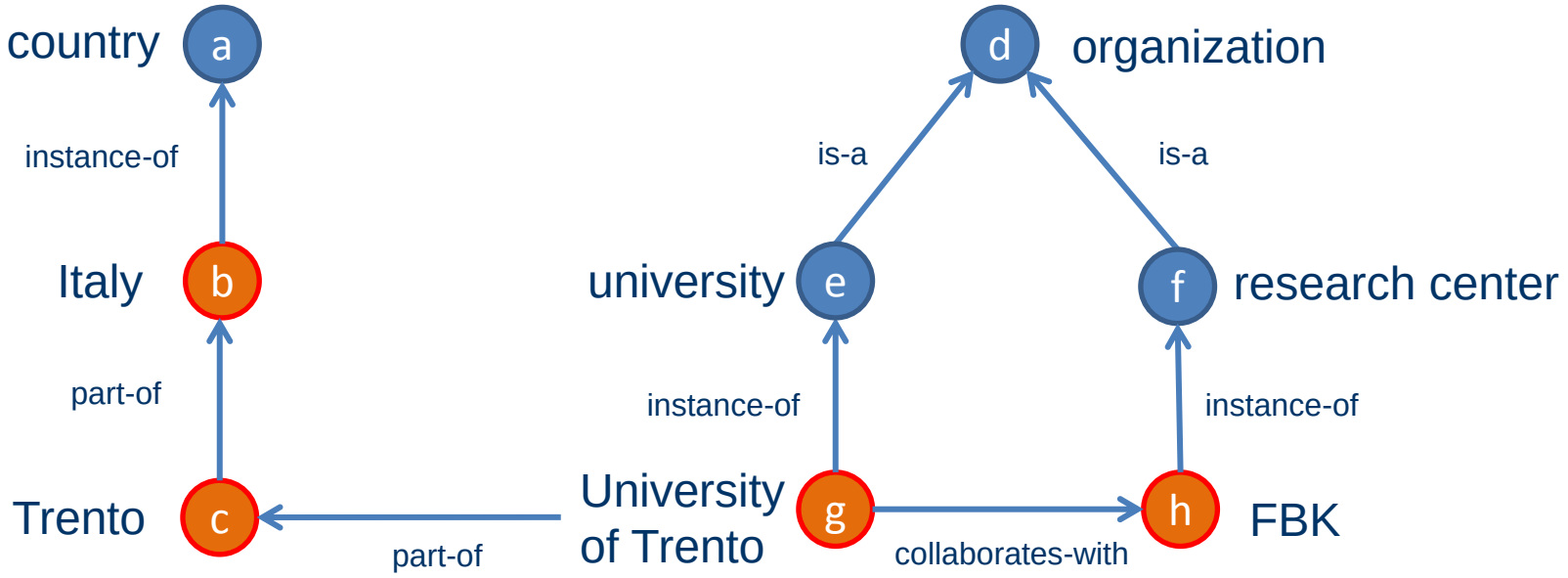
maltese@disi.unitn.it

GeoWordNet open source:

http://geowordnet.semanticmatching.org/

Extra slides

Descriptive ontologies: how to make them formal



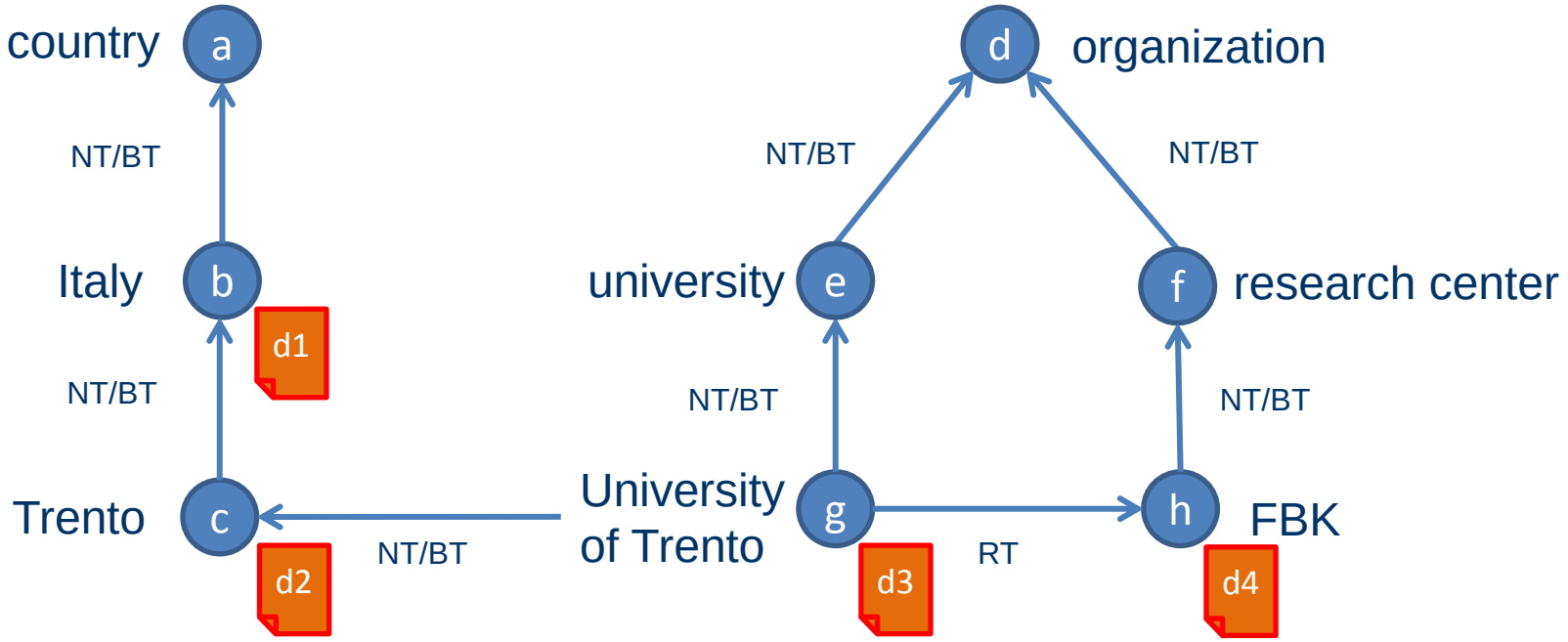
university ⊑ organization

university(UniversityOfTrento)

collaborates(UniversityOfTrento, FBK)

…

Classification ontologies: how to make them formal



university ⊑ organization

Trento ⊑ Italy

Italy(d1)

…

Limitations of RDF and SKOS

RDF	SKOS
No support for disjointness	**No support for disjointness**
Classes can be treated as instances	**No distinction between classes and instances (we cannot represent documents)**
Transitivity of relations cannot be enforced at the level of entities	**We can define non-transitive NT/BT**

Is b ≡ d ?