Translation Coherence Knowledge Engineering project

LORENZO MARIO AMOROSA, MICHELE IANNELLO, ANDREA LAVISTA
MASTER DEGREE IN ARTIFICIAL INTELLIGENCE — UNIVERSITY OF BOLOGNA

Overview
and preliminary steps

- **Translation coherence**: understanding linguistic **variations** occurring throughout translations across different languages
- **Method**: compare **back-and-forth** translations (from/to English)
- **Knowledge Engineering** tools: **FRED** (machine-reading), **Protégé** (ontologies designing), **Virtuoso** (ontologies querying), **LodView** (ontologies browsing), **LODE** (documentation browsing), **RDFLib** (Python library)
- Other tools: Docker, **NLP** Python libraries (spaCy, NLTK, WordNet®)
- **Data sources**:
  - **Europarl** (*European Parliament Proceedings Parallel Corpus 1996-2011*) for *text sources*
  - **DeepL Translate** for *translations*
  - **FRED** for *machine-reading*

Custom ontology alignment

- Experiments with **LIMES** (*Link Discovery Framework for Metric Spaces*)
  - Allows for **ontology alignment**
  - Not suitable for our purposes
- **Translation coherence**
  - **Goals**
    - Searching for recurrent **semantic patterns**
    - **Classification** of semantic differences
  - **Methods**
    - (Exact and inexact) **Graph Matching**
    - (Custom) **ontology alignment**

Custom approach towards semantic analysis
Design and engineering

- Draw inspiration from **eXtreme Design methodology** principles:
  - Use of **competency questions**
  - Identification of *Ontology Design Patterns* (**ODPs**)
  - Ontologies *testing* through **SPARQL** queries

Custom approach towards semantic analysis
Competency question – Example

- *What are the differences between the two ontologies?*

```
PREFIX tc:<https://w3id.org/stlab/ke/amiala/translation_coherence/>
  SELECT ?s ?p ?o
  WHERE{
    {?s tc:different ?o} UNION
    {?s tc:synonymy ?o} UNION
    {?s tc:differentHierarchy ?o} UNION
    {?s tc:similarHierarchy ?o} UNION
    {?s tc:differentExpression ?o}
    ?s ?p ?o .
  }
```

Custom approach towards semantic analysis
Competency question – Example

- *Which are the synonymy relations in the A-Box of the two ontologies?*

```
PREFIX tc:<https://w3id.org/stlab/ke/amiala/translation_coherence/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

  SELECT ?s ?o
  WHERE{
    ?s tc:synonymy ?o ;
    rdf:type ?type
    FILTER(?type != tc:ClassConcept)
  }
```

Custom approach towards semantic analysis
Workflow

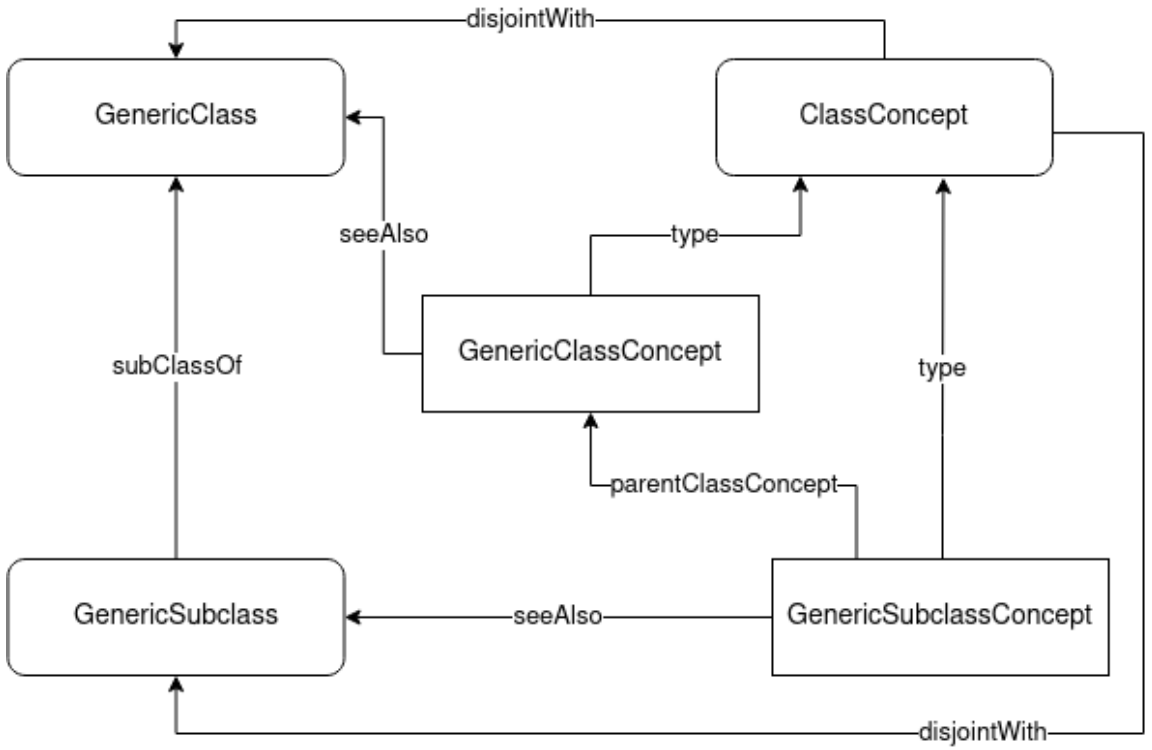
- **Sample sentences** were gathered from **Europarl** corpus (language: *English*)
- **DeepL** was used to **translate** English sentences to *German/Italian/Chinese* and back to *English*
- **Translations** were submitted to FRED, producing knowledge graphs as output (encoded using Turtle syntax)
- **Knowledge graphs (KGs)** were pairwise **manually inspected** (original vs double translations) and **automatically processed** with the help of Python libraries `RDFLib`, `spaCy`, `NLTK`
- A **vocabulary** was designed and engineered to **shape the representation** of the comparisons
- **Output** ontologies (`.owl`) representing the **comparisons** were produced (one for each pair) and uploaded onto the GitHub repository of the project
- The services **Virtuoso**, **LODE**, **LodView** were set up (within a Docker container) to allow **querying the results** and to **ease the browsing** of the ontologies

Custom approach towards semantic analysis
Building knowledge graphs

- **Input** KGs:
  - A **pair** of KGs is selected
  - KGs are **cleaned** from unnecessary information generated by Fred (such as *textual reference offset*) and **enriched** with labels (`rdfs:label`)
  - **Lemmatization** is performed on all the labels to allow for *semantic* reasoning
- **Output** KGs:
  - The **Intensional reification ODP** is deployed to allow making statements about classes

Apply Intensional Reification ODP

- **Intensional Reification ODP (inspired from w3.org)**



Custom approach towards semantic analysis
Comparing knowledge graphs

- **Core strategy**:
  - Identification of **strong equivalences** across the KGs as *starting points*
    - Pairs of nodes sharing **at least 3 identical triples**
    - Pairs of classes with the **same name**
    - Common ground resources used by Fred (from boxing, quantifiers, etc.)
  - Iterative **propagation** of equivalences to **whole KGs** beginning with starting points
  - Detection of matching **recurrent subgraph structures** (*patterns*) across KGs
- Pseudocode:

```
frontiers = find_starting_points(graph1, graph2)
while(frontiers):
    while(frontiers):
        propagate_equivalences(graph1, graph2, frontiers)
        apply_safe_patterns(graph1, graph2, frontiers)
    apply_unsafe_patterns(graph1, graph2, frontiers)
```

Custom approach towards semantic analysis
Patterns in the inner loop

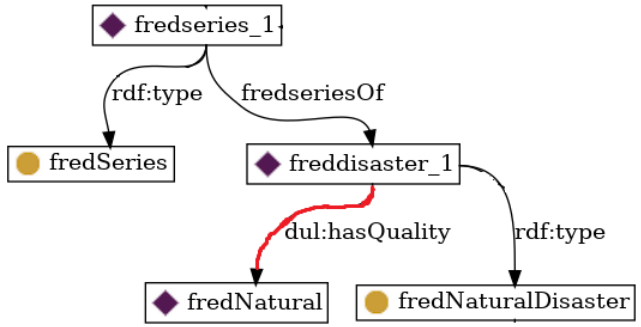
- **Find equivalence relations**
  - Pairs of individuals with **same lemma and respectively linked** to nodes forming a pair in the list *frontiers* with the **same predicate**

- **Find synonymy relations**
  - Pairs of individuals with **synonym lemmas** (or classes with **synonym names**) and respectively **linked** to nodes forming a pair in the list *frontiers* with the **same predicate**
  - **Synonymy** conditions of a pair of nodes based on **WordNet®** (at least one should hold):
    - they share at least one **synset** (synsets are "*sets of cognitive synonyms, each expressing a distinct concept*")
    - their **wup** (*Wu-Palmer*) **similarity** is at least 0.85
    - their **path similarity** is at least 0.45

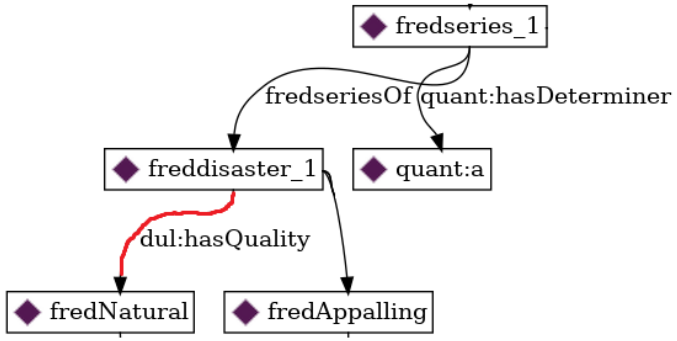
Custom approach towards semantic analysis

Patterns in the inner loop – Examples

- **Find equivalence relations**

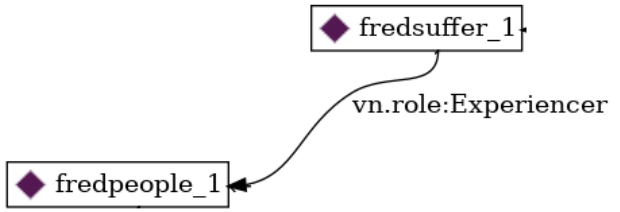


en_sentence2 subgraph

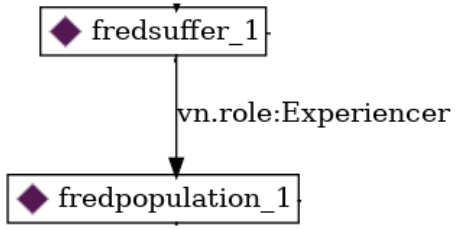


en_it_en_sentence2 subgraph

- **Find synonymy relations**



en_sentence2 subgraph



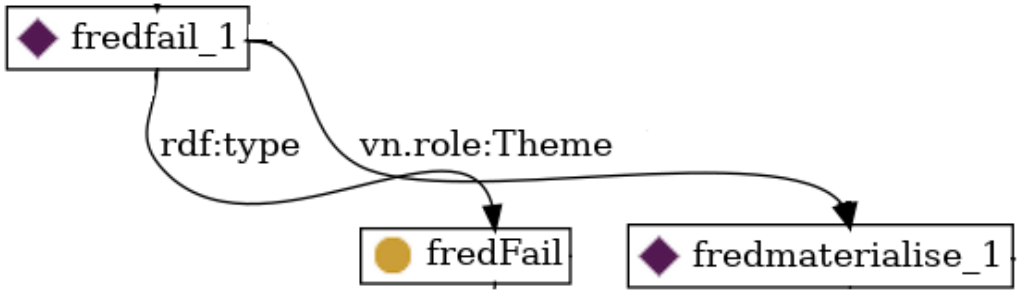
en_it_en_sentence2 subgraph

Custom approach towards semantic analysis
Patterns in the inner loop

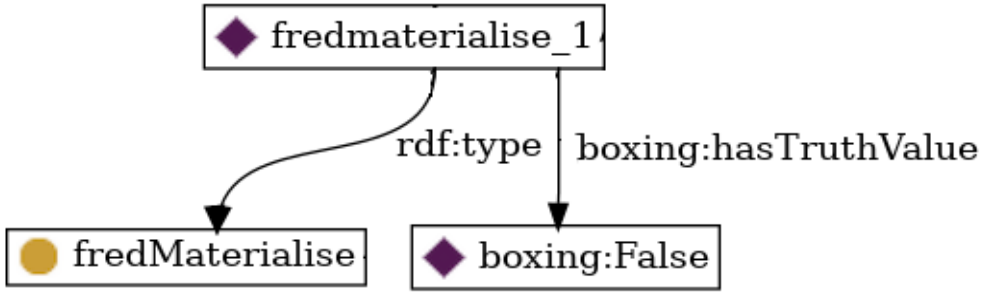
- **Negative verbs**
  - **Textual** expression variations (e.g. *fail to materialise* rendered as *didn't materialise/appear*)
  - Represented through the **N-ary relation Logical ODP** in the result graphs

Custom approach towards semantic analysis
Patterns in the inner loop – Examples

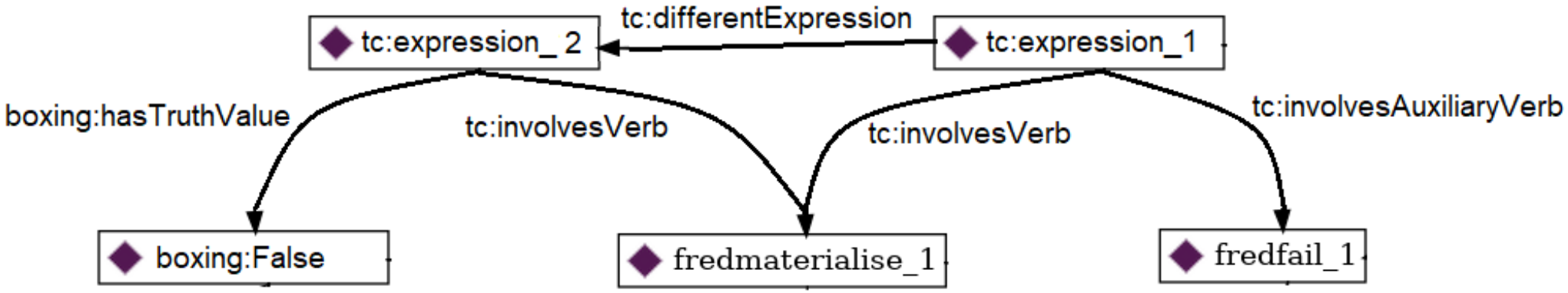
- **Negative verbs**



en_sentence2 subgraph



en_it_en_sentence2 subgraph



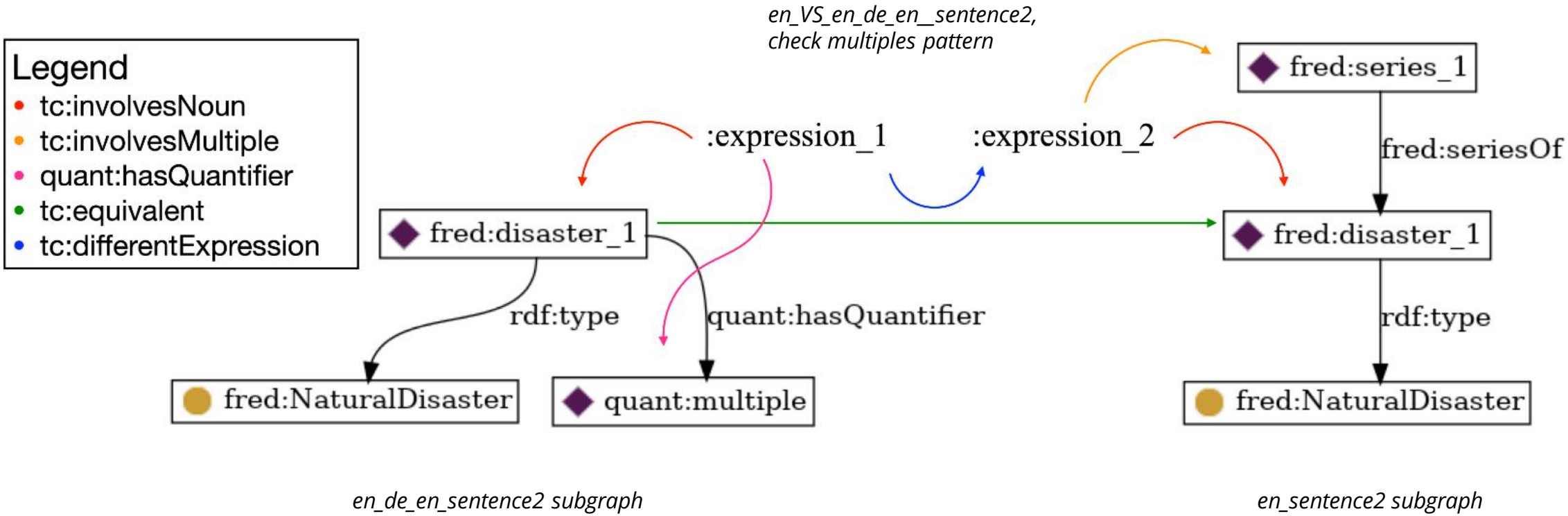
en_VS_en_it_en__sentence2, negative verbs pattern

Custom approach towards semantic analysis
Patterns in the inner loop

- **Quantifiers analysis**
  - **Representation** expression variations (example follows)
  - Represented through the **N-ary relation Logical ODP** in the result graphs

Custom approach towards semantic analysis
Patterns in the inner loop – Examples

- **Quantifiers analysis**



Custom approach towards semantic analysis
Patterns in the inner loop

- **Class-subclass analysis**

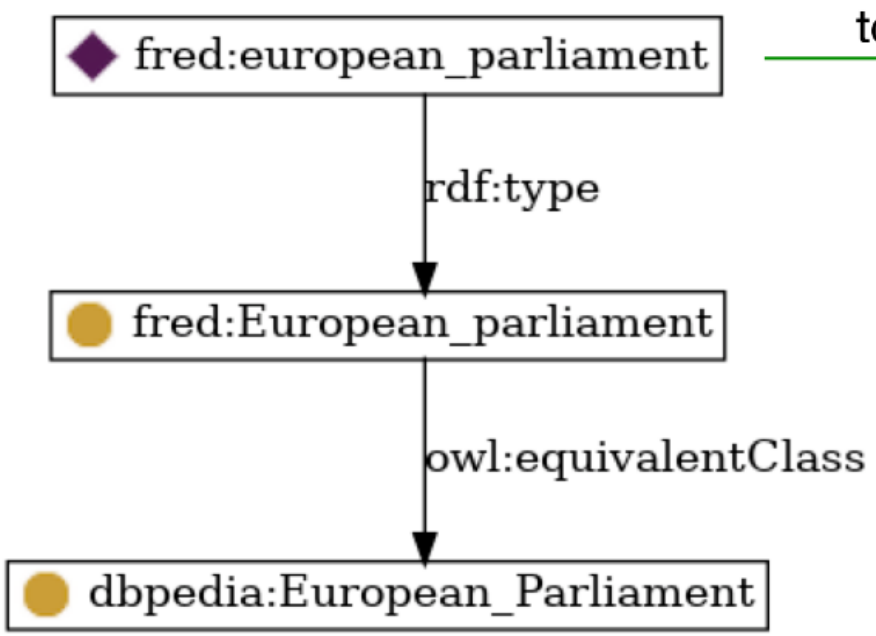
  Pairs of nodes in *frontiers* exhibiting one of the following properties:

  - **Individual-class fix**: nodes belong to a structure of equivalences wherein a certain resource is present in both KGs, once as individual and once as a class
    - The **individual** resource and the (eventual) instance of the **class** resource are declared *equivalent*
  - **Equivalence/synonymy** propagation: nodes are members of **hierarchies of classes** which are found either **pairwise equivalent** or **synonym**
    - The pairs are respectively linked through the corresponding predicate (*equivalent* or *synonymy*)
  - **Difference** propagation/**Hierarchy** generation: nodes are classified as *different* or belong to **non-straightforwardly relatable hierarchies of classes**
    - Two **N-ary relation** of class *Hierarchy* are created and linked through the corresponding predicate *(differentHierarchy* or *similarHierarchy)*

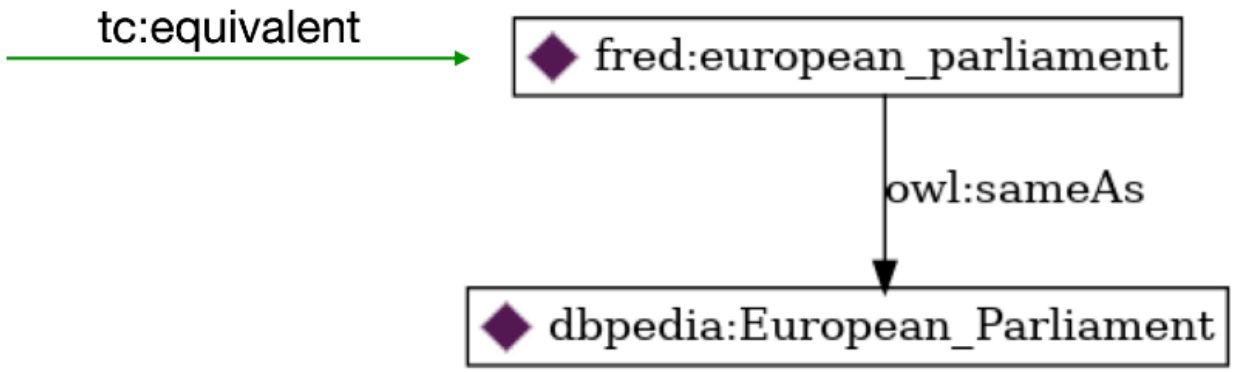
Custom approach towards semantic analysis

Patterns in the inner loop – Examples

- **Class-subclass analysis**
  - **Individual-class fix**



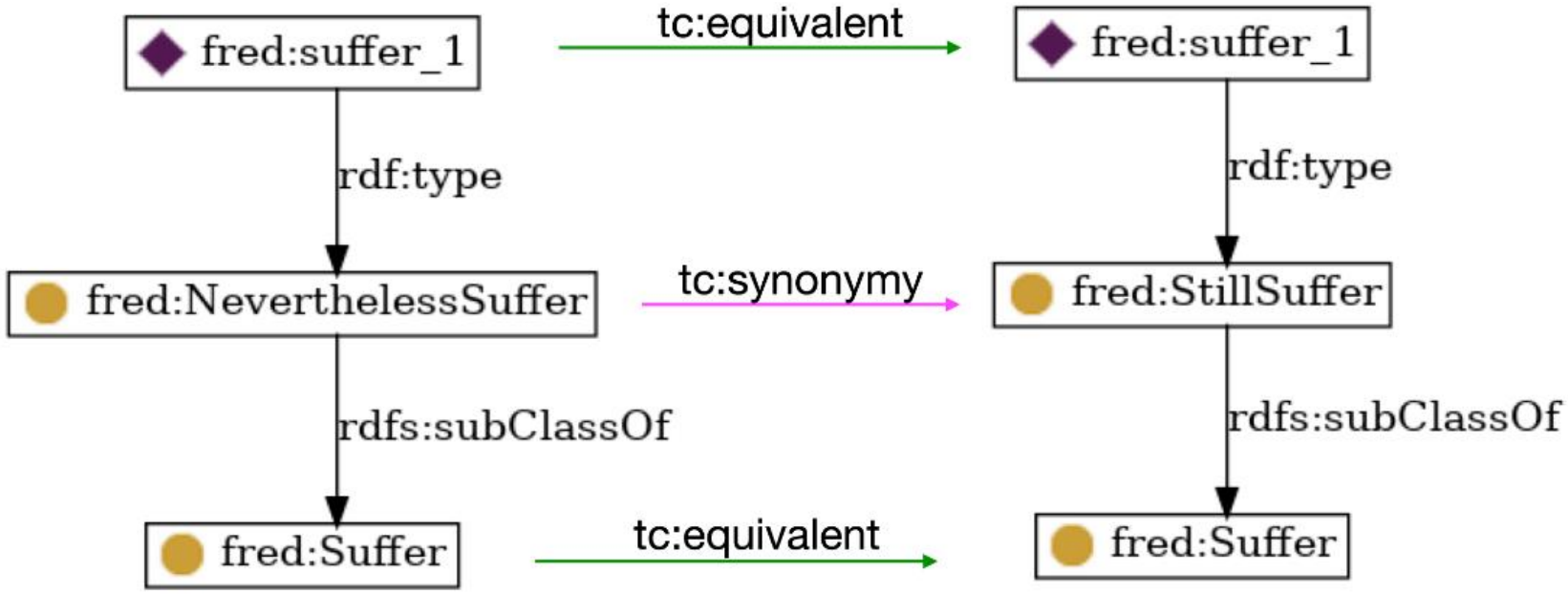
en_sentence1 subgraph



en_it_en_sentence1 subgraph

Custom approach towards semantic analysis
Patterns in the inner loop – Examples

- **Class-subclass analysis**
  - **Equivalence/synonymy propagation**

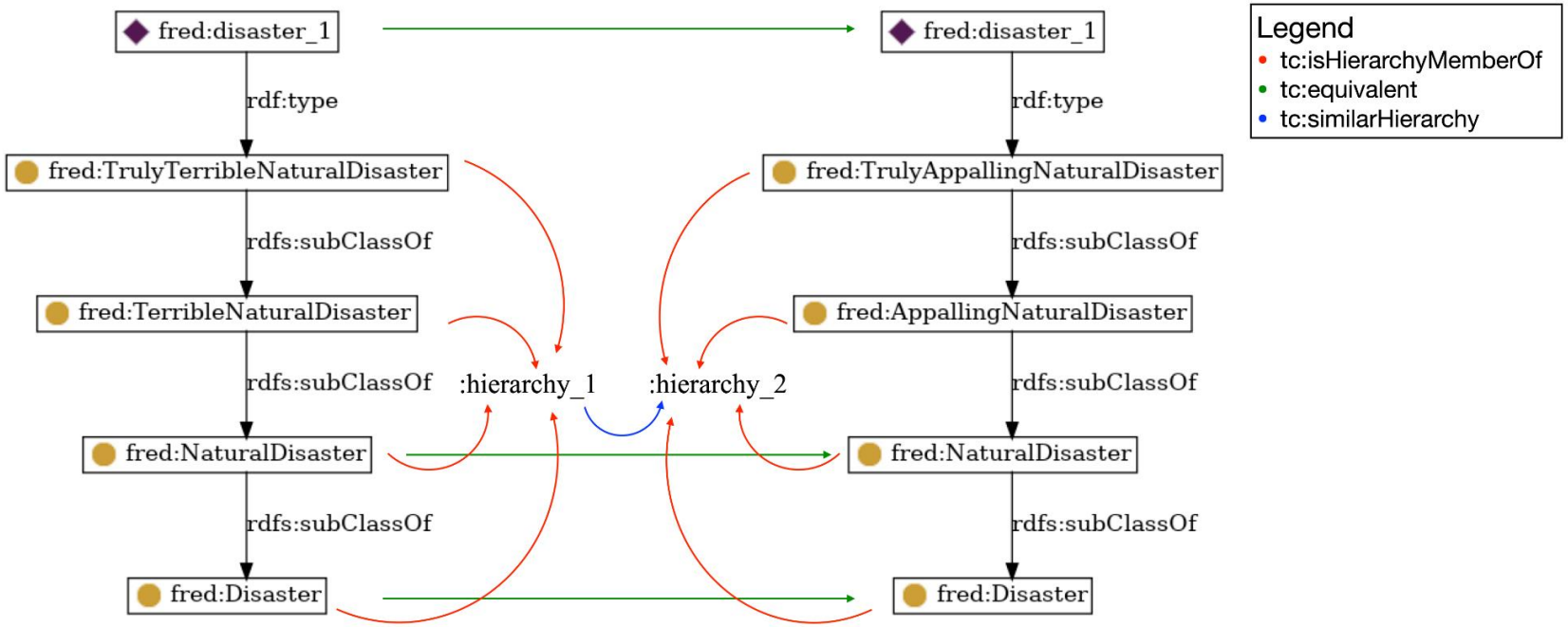


en_de_en_sentence2 subgraph

en_cn_en_sentence2 subgraph

Custom approach towards semantic analysis
Patterns in the inner loop – Examples

- **Class-subclass analysis**
  - **Hierarchy generation**



en_cn_en_sentence2 subgraph

en_it_en_sentence2 subgraph

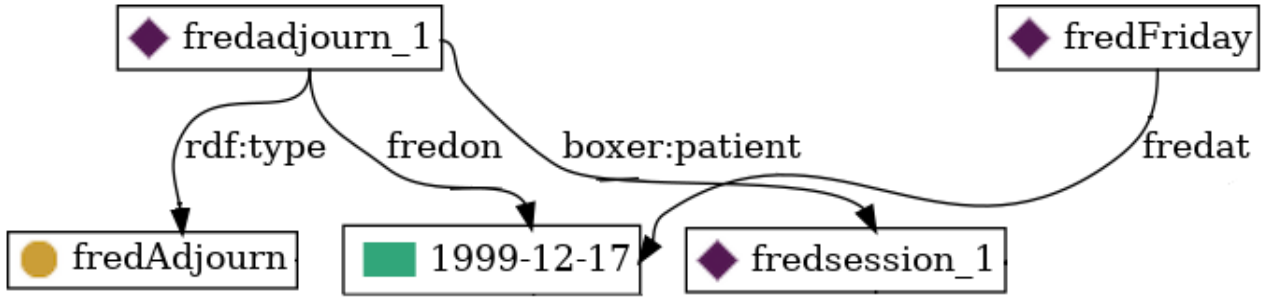
Custom approach towards semantic analysis
Patterns in the outer loop

- **Find binary difference relations**
  - Classify nodes which are not equivalent, but that accomplish the **same role in the input ontologies**
- **Find synonymy classes**
  - Mark as equivalent nodes all those **classes whose lemmas are synonyms**, to have more starting points

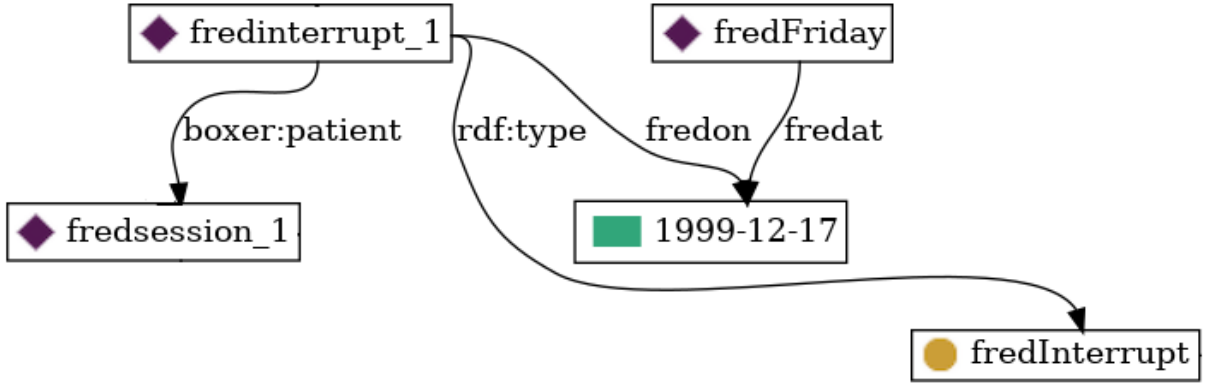
Custom approach towards semantic analysis

Patterns in the outer loop – Examples

- **Find binary difference relations**



en_sentence1 subgraph



en_it_en_sentence1 subgraph

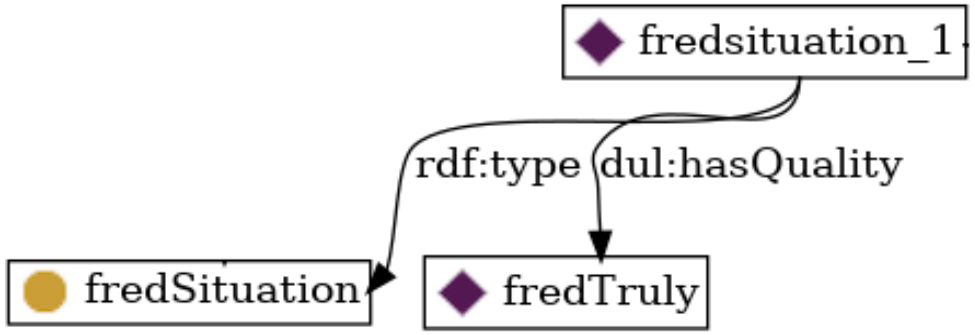
Custom approach towards semantic analysis
Patterns outside the loops

- **All different relations**
  - Nodes which belong to both KGs and that **don't share any triple across KGs**
- **Only in one graph**
  - Nodes which belong to **only one** of the input **KGs**

Custom approach towards semantic analysis

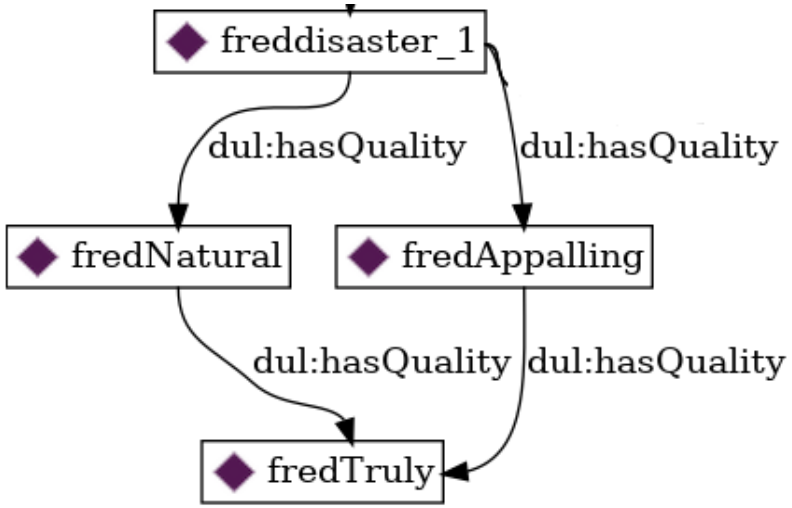
Patterns outside the loops – Examples

- **All different relations**



en_sentence2 subgraph

"a series of natural disasters that truly were dreadful"



en_it_en_sentence2 subgraph

"a series of truly appalling natural disasters"

Custom approach towards semantic analysis
Exporting the knowledge graphs

- ~~**Permanent** IRIs~~
  - All the classes and individuals are published using the namespace:

    `https://w3id.org/stlab/ke/amiala/translation_coherence/`
- **OWL/XML Syntax**
  - All the ontologies (input ones, result graphs) are serialized using the **OWL format**

Deployment

- **GitHub repository** to host code scripts, ontologies and documentation
- **Docker container**

  Provided as a subfolder within the repository, it allows to run the following services

  - **Virtuoso**
    - Ontologies **querying**: the SPARQL endpoint allows retrieving data stored in the result ontologies
  - **LODE**
    - **Documentation** browsing: the service provides docs about ontologies (as HTML pages)
  - **LodView**
    - Ontologies **browsing**: the service provides HTML representations of ontologies and resources

Thank you for your attention

References

- *Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, Misael Mongiovì.* **"Semantic Web Machine Reading with FRED"**. Semantic Web Journal 8(6):873-893, 2017.
- *Philipp Koehn*, **"Europarl: A Parallel Corpus for Statistical Machine Translation"**, MT Summit 2005
- *Axel-Cyrille Ngonga Ngomo, Mohamed Ahmed Sherif, Kleanthi Georgala, Mofeed Hassan, Kevin Dreßler, Klaus Lyko, Daniel Obraczka, Tommaso Soru*. **"LIMES - A Framework for Link Discovery on the Semantic Web".** KI-Künstliche Intelligenz, German Journal of Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V. 2021.
- *Carl Boettiger*. **"rdflib: A high level wrapper around the redland package for common rdf applications (Version 0.1.0)**." Zenodo 2018.
- *Presutti V., Daga E., Gangemi A., Blomqvist E.* **"eXtreme Design with Content Ontology Design Patterns."** WOP (2009).
- *Musen M.A.* "**The Protégé project: A look back and a look forward."** AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), June 2015.
- *Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart*, **"Exploring network structure, dynamics, and function using NetworkX"**, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008
- *Honnibal et al.* **"spaCy: Industrial-strength Natural Language Processing in Python"**. Zenodo 2020.

References

- *George A. Miller (1995).* **"WordNet: A Lexical Database for English."** Communications of the ACM Vol. 38, No. 11: 39-41.
- *Christiane Fellbaum (1998, ed.)* **"WordNet: An Electronic Lexical Database."** Cambridge, MA: MIT Press.
- *Bird, Steven, Edward Loper and Ewan Klein (2009),* **"Natural Language Processing with Python."** O'Reilly Media Inc.
- *Jeffrey Pennington, Richard Socher, and Christopher D. Manning.* **"GloVe: Global Vectors for Word Representation."** 2014.
- *Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z.* **"DBpedia: A Nucleus for a Web of Open Data."** In: Aberer K. et al. (eds) The Semantic Web. ISWC 2007, ASWC 2007. Lecture Notes in Computer Science, vol 4825. Springer, Berlin, Heidelberg. (2007)
- *Gangemi A.* **"Ontology Design Patterns for Semantic Web Content."** In: Gil Y., Motta E., Benjamins V.R., Musen M.A. (eds) The Semantic Web – ISWC 2005. ISWC 2005. Lecture Notes in Computer Science, vol 3729. Springer, Berlin, Heidelberg.
- *Merkel D.* **"Docker: lightweight linux containers for consistent development and deployment."** Linux journal. 2014;2014(239):2.
- *Peroni S., Shotton D.M., Vitali F.* **"Making Ontology Documentation with LODE."** In Proceedings of the I-SEMANTICS 2012 Posters & Demonstrations Track, Graz, Austria, September 5-7, 2012, 63–67, 2012.
- *Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio.* **"LodLive, exploring the web of data."** In Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS'12). ACM, New York, 197--200. (2012)
- *Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio.* **"LodView."** 2014
- Virtuoso Open-Source Edition, link