CITY
UNIVERSITY OF LONDON
EST 1894

Applications to Data Science

Ernesto Jiménez-Ruiz
Lecturer in Artificial Intelligence

Before we start...

Some important information

- Course work submission reminder:
  - Sunday, 6 March 2022, 5:00 PM.
  - Submit files individually as mentioned in the instructions (*i.e.*, *myOnto.ttl, myOnto.owl, myVideo.mp4 and myOntoCode.pdf*).
  - If video is $> 200Mb$ then upload it into Google/One drive and upload in moodle a text file with the link to the video (*i.e.*, *myVideoLink.txt*).
- Drop-in sessions during reading week.
  - Tuesday: 1-2:30pm (online). Using wonder.me: `https://bit.ly/in3067-inm713-online-lab`
  - Thursday: 3-4pm (as usual in Zoom) - Coursework Part 2 discussion

Where are we? Module organization (tentative)

✓ Introduction.
✓ OWL (2) ontology language. Focus on modelling.
✓ RDF-based knowledge graphs.
✓ SPARQL 1.0 Query Language.

5. **Applications to Data Science**. From tabular data to KG. (Today)
6. RDFS Semantics and OWL 2 profiles.
7. SPARQL 1.1 and Graph Database solutions.
8. Ontology Alignment.
9. Ontology Embeddings and Machine Learning.
10. Other topics. *Possibly RML mappings, SHACL constraint language...*

Data Science Bottleneck

Data Science Bottleneck

Big Data Borat
@BigDataBorat

Follow

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

Data Science Bottleneck

Big Data Borat
@BigDataBorat

Follow

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

One of the main problems is the lack of understanding of the data and the domain.

Contribution of Semantics in Data Wrangling Challenges

- *Data parsing*, e.g. converting csv's or tables.
- (+++)*Data dictionary*: basic types and semantic types.
- (++)*Data integration* from multiple sources (foreign key discovery).
- (++)*Entity resolution*: duplication and record linkage.
- (+)*Format variability*: e.g. for dates and names.
- (+)*Structural variability* in the data.
- (++)Identifying and repairing *missing data*.
- (+)*Anomaly detection* and repair.
- (+++)**Metadata/contextual information**. (Semantic) data governance.

AIDA Project: `https://www.turing.ac.uk/research/research-projects/artificial-intelligence-data-analytics-aida`

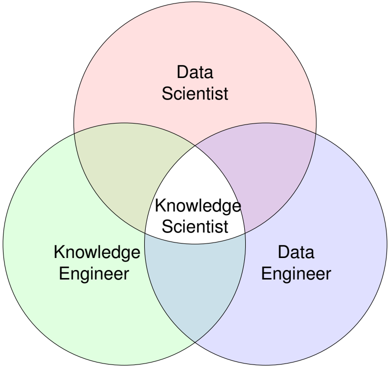
Semantic Understanding of Tabular Data

- **Semi-automatic** process.
- Key for an **enhanced transformation** to RDF triples.
- But also for other tasks with independence of a final KG creation.
  - Tabular data in the form of CSV files is the common input format in a **data analytics pipeline**.
  - The **lack of semantics and context in datasets** hinders their usability.
  - Gaining **semantic understanding** will be very valuable for data integration, data cleaning, data mining, machine learning and knowledge discovery tasks.

The Knowledge Scientist

The Knowledge Scientist (i)

- **Data Engineer:** harnesses and collects data.
- **Data Scientist:** draws value from data.
- **Knowledge Engineer:** encodes domain expertise.
- **Knowledge Scientist:** adds context to the data to make it more useful, clean, reliable and ready to be used.



The Knowledge Scientist (ii)

– Bridges the data and the **business requirements**/questions.

– Outputs a data model (*i.e.*, **knowledge graph**): how business users see the world.

– Drives a **semantic-lifting** of the data (from Data Engineers to Data Scientists)

– Relies on **Semantic Web technology** and skills (*e.g.*, ontology modelling, data/knowledge integration, graph databases)

George Fletcher and others. **Knowledge Scientists: Unlocking the data-driven organization**. 2020

The Knowledge Scientist: DS perspective

- Domain understanding.
- Designing & developing ontologies and KGs together with domain experts.
- Developing data processing pipelines to drive the semantic shifting of the data.
- Application of data analytics and ML with KGs.

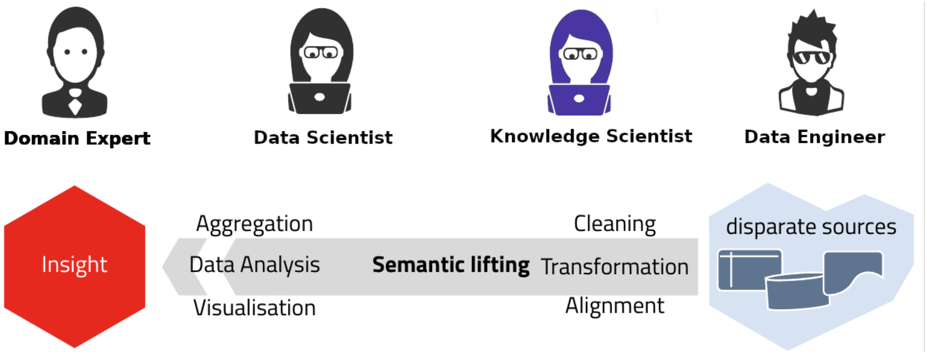
George Fletcher and others. **Knowledge Scientists: Unlocking the data-driven organization**. 2020
Stratos Kontopoulos. **Semantic AI & the Role of the Knowledge Scientist**. 2021

The Knowledge Scientist: SE/CS perspective

- Domain understanding.
- Deploying the semantic infrastructure (*e.g.*, triplestores, reasoning, query formulation, required APIs).
- Integration with other components.

George Fletcher and others. **Knowledge Scientists: Unlocking the data-driven organization**. 2020
Stratos Kontopoulos. **Semantic AI & the Role of the Knowledge Scientist**. 2021

The Knowledge Scientist: overview



Adapted from: SIRIUS Centre for Scalable Data Access, `https://sirius-labs.no/`

Why Ontologies and Graphs of Knowledge?

Graph(s) of Knowledge / Knowledge Graphs

- Semantic Web in more **controlled scenarios**, *e.g.*,
  - **Integrate and orchestrate** data within an organisation
  - Enterprise data as a knowledge graph to **drive products** and make them more "**intelligent**"

Graph(s) of Knowledge / Knowledge Graphs

- Semantic Web in more **controlled scenarios**, *e.g.*,
  - **Integrate and orchestrate** data within an organisation
  - Enterprise data as a knowledge graph to **drive products** and make them more “**intelligent**”

- **Not new**:
  - Graph data models extensively studied in AI…
  - …but Google has relaunched the interest on **KGs in industry**

Graph(s) of Knowledge / Knowledge Graphs

- Semantic Web in more **controlled scenarios**, *e.g.*,
  - **Integrate and orchestrate** data within an organisation
  - Enterprise data as a knowledge graph to **drive products** and make them more "**intelligent**"
- **Not new**:
  - Graph data models extensively studied in AI…
  - …but Google has relaunched the interest on **KGs in industry**
- Availability of **mature** Semantic Web **technology**
  - Query engines
  - Modelling languages
  - Reasoning

Ontologies and Knowledge Graphs

– Core idea of knowledge graphs is the enhancement of the graph data model with...
  – “...an **abstract symbolic representations** of a domain expressed in a formal language”

– In this module: **OWL-layered RDF-based knowledge graphs**

Aidan Hogan and others. **Knowledge Graphs**. 2021 `https://kgbook.org/`.
Pim Borst, Hans Akkermans, and Jan Top. **Engineering ontologies**, 1999.

Why Ontologies and KGSs?

– Independence of logical/physical schema: **domain model**

– Vocabulary closer to domain experts: **more user-friendly**

– Incomplete and semi-structured data: **flexibility**

– Integration of heterogeneous sources: **unified view**

♠ They can complement tabular data not necessarily substitute.

Why Ontologies and KGs? (Experiences from Industry)

- Knowledge graphs are **not complex**: they are the simplest way to layer human expertise onto stored data.
- Investing in knowledge graphs is **resilient**: an initial investment that pays off by allowing companies to pre-prepare for unanticipated and unfolding use cases
- **Unifying logical representation** for data an semantics to get out the data science enterprise mess.

Juan Sequeda and Ora Lassila. **Designing and Building Enterprise Knowledge Graphs**. 2021
Ora Lassila. **On the broad applicability of Semantic Web technologies**. (COST DKG Talk Series. 2021
https://www.youtube.com/watch?v=f9wautaqWUs
Juan Sequeda and Ora Lassila. https://watch.knowledgegraph.tech/knowledge-espresso/videos/knowledge-espresso-with-ora-lassila-and-juan-sequeda

FAIR principles and 5-star data

Why Ontologies and KGs?

5-star Data

- ⋆ **OL**: make your data available on the Web (in any format) under an open license.

Tim Berners-Lee. **5 ⋆ data:** `https://5stardata.info/en/`

5-star Data

- ⋆ **OL**: make your data available on the Web (in any format) under an open license.
- ⋆⋆ **RE**: make the data machine readable (excel instead of an scanned image).

Tim Berners-Lee. **5 ⋆ data:** `https://5stardata.info/en/`

5-star Data

- ⋆ **OL**: make your data available on the Web (in any format) under an open license.
- ⋆⋆ **RE**: make the data machine readable (excel instead of an scanned image).
- ⋆⋆⋆ **OF**: use a non proprietary open format (*e.g.*, CSV).

Tim Berners-Lee. **5 ⋆ data:** `https://5stardata.info/en/`

5-star Data

- ⋆ **OL**: make your data available on the Web (in any format) under an open license.
- ⋆⋆ **RE**: make the data machine readable (excel instead of an scanned image).
- ⋆⋆⋆ **OF**: use a non proprietary open format (*e.g.*, CSV).
- ⋆⋆⋆⋆ **URI**: use URIs instead of strings (RDF).

Tim Berners-Lee. **5 ⋆ data:** `https://5stardata.info/en/`

5-star Data

- $\star$ **OL**: make your data available on the Web (in any format) under an open license.
- $\star\star$ **RE**: make the data machine readable (excel instead of an scanned image).
- $\star\star\star$ **OF**: use a non proprietary open format (*e.g.*, CSV).
- $\star\star\star\star$ **URI**: use URIs instead of strings (RDF).
- $\star\star\star\star\star$ **LOD**: link your data to other data to provide context.

Tim Berners-Lee. **5 $\star$ data:** `https://5stardata.info/en/`

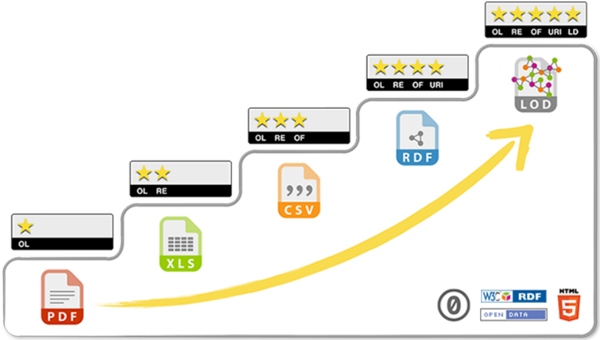
5-star Data

- ⋆ **OL**: make your data available on the Web (in any format) under an open license.
- ⋆⋆ **RE**: make the data machine readable (excel instead of an scanned image).
- ⋆⋆⋆ **OF**: use a non proprietary open format (*e.g.*, CSV).
- ⋆⋆⋆⋆ **URI**: use URIs instead of strings (RDF).
- ⋆⋆⋆⋆⋆ **LOD**: link your data to other data to provide context.

♠ This also applies within a company (intranet), not only for the Web. Ideally with an OL, but at least data accessible by everyone in the company.

Tim Berners-Lee. **5 ⋆ data:** `https://5stardata.info/en/`

5-star Data



♠ **LOD**: Linked Open Data.

Tim Berners-Lee. **5 ⋆ data:** `https://5stardata.info/en/`

5-star Data: Technical challenges:

- How to **expose** data (*e.g.*, databases, csv files) as knowledge graphs?
- How to **create** (or reuse) and use (abstract) **knowledge** (*i.e.*, *Ontologies*)?
- How to **align** different knowledge graphs? ♠
- How to check **consistency and trust** of the data and knowledge? ♠

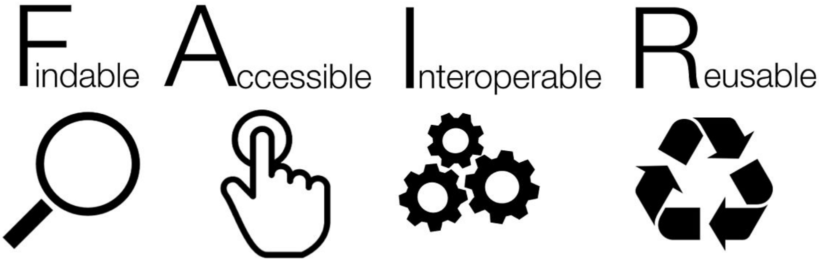
♠ Better with things than with strings

5-star Data: Technical challenges:

- How to **expose** data (*e.g.*, databases, csv files) as knowledge graphs?
  - *RDF (Week 3) and Today's session - 4-5 ⋆ data*
- How to **create** (or reuse) and use (abstract) **knowledge** (*i.e.*, *Ontologies*)?
  - *OWL (Week 2)*
- How to **align** different knowledge graphs? ♠
  - *Ontology Alignment (Week 8) - 5 ⋆ data*
- How to check **consistency and trust** of the data and knowledge? ♠
  - *Reasoning with RDFS and OWL (Week 7) - 6 ⋆ data?*

♠ Better with things than with strings

FAIR Data Principles (i)



♠ For for machines and people.

The FAIR Guiding Principles for scientific data management and stewardship. Nature Scientific Data 2016.
`https://www.go-fair.org/fair-principles/`

FAIR Data Principles (ii)

- **Findable**: (meta)data is assigned an unique identifier and data is described with rich metadata.
- **Accessible**: (meta)data can be accessed via a known protocol. ($\star$)
- **Interoperable**: meta(data) uses a formal (knowledge representation) language.
- **Reusable**: well described (meta)data with rich provenance.

($\star$) Not necessarily open. (FAIR $\neq$ OPEN)

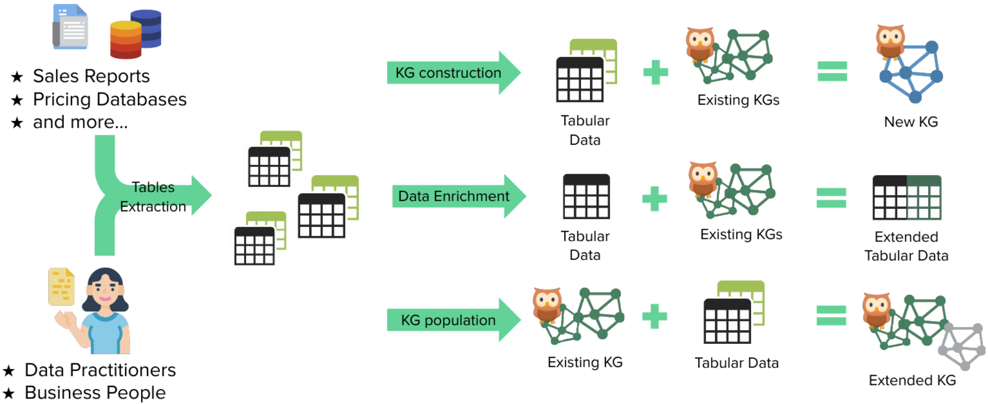
FAIR Data Principles (iii)

- FAIR data is more valuable, easier to find and combine thanks to unique identifiers and a formal shared knowledge representation.
- “KGs must be in want of FAIR data. And FAIR data is in want of KGs”.

Carole Golbe. FAIRy stories: the FAIR Data principles in theory and in practice.
`https://www.slideshare.net/carolegoble/fairy-stories-the-fair-data-principles-in-theory-and-in-practice`

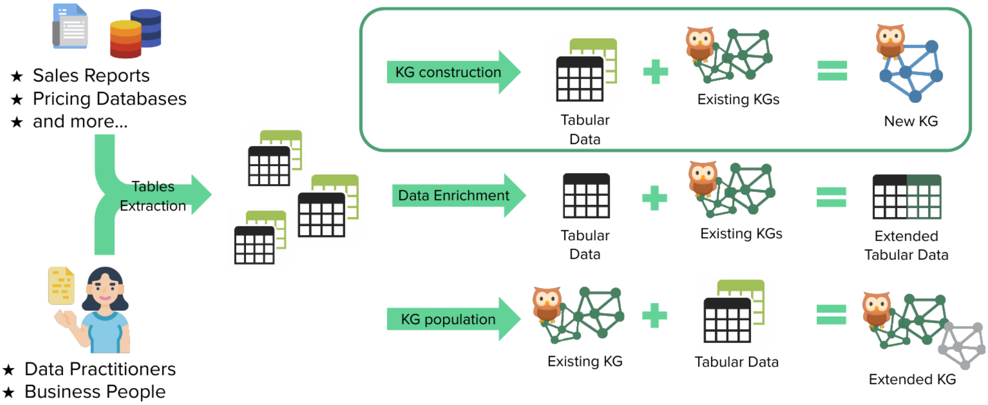
From (Tabular) Data to Knowledge Graphs:
Towards 5 $\star$ and FAIR data

Semantic Table Understanding/Interpretation



Vincenzo Cutrona. **Why Table Understanding Matters**. See last year invited speakers.

Semantic Table Understanding/Interpretation



Vincenzo Cutrona. **Why Table Understanding Matters**. See last year invited speakers.

Exposing data as an RDF-based Knowledge Graph

- ✓ **End-users’ friendly access** to “unfriendly” tabular data.
- ✓ **Pay as you go** (modular) data integration via mappings.

Exposing data as an RDF-based Knowledge Graph

✓ **End-users' friendly access** to "unfriendly" tabular data.

✓ **Pay as you go** (modular) data integration via mappings.

– **Option 1**: **Virtual exposure of data** (OBDA)

  ✓ Data remains in its original format.

  ✗ Typically only over relational databases.

Exposing data as an RDF-based Knowledge Graph

✓ **End-users' friendly access** to "unfriendly" tabular data.

✓ **Pay as you go** (modular) data integration via mappings.

– **Option 1**: **Virtual exposure of data** (OBDA)
  - ✓ Data remains in its original format.
  - ✗ Typically only over relational databases.

– **Option 2**: **Data Export/Materialization**
  - ✓ Easy to exchange data (RDF).
  - ✓ Integration of data in disparate formats.
  - ✗ Data replication.
  - – Due to size or privacy it may not be possible to export the data.

Exposing data as an RDF-based Knowledge Graph

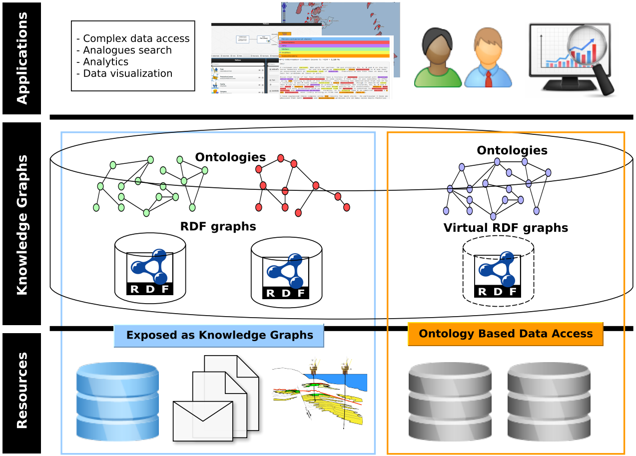
✓ **End-users' friendly access** to "unfriendly" tabular data.

✓ **Pay as you go** (modular) data integration via mappings.

– **Option 1**: **Virtual exposure of data** (OBDA)
  - ✓ Data remains in its original format.
  - ✗ Typically only over relational databases.

– **Option 2**: **Data Export/Materialization** (In this module)
  - ✓ Easy to exchange data (RDF).
  - ✓ Integration of data in disparate formats.
  - ✗ Data replication.
  - – Due to size or privacy it may not be possible to export the data.

Exposing data as RDF: Architecture

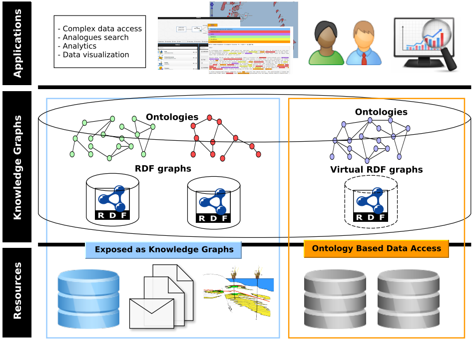


Exposing data as RDF: Ingredients

– **Ontology vocabulary**. Custom and/or given by a public KG.

– **Mappings**. Define a transformation from the tabular data to RDF data.

– **Ontology Axioms** (optional) - ♠



♠ Ernesto Jimenez-Ruiz and others. **BootOX: Practical Mapping of RDBs to OWL 2**. ISWC 2015

Exposing data as RDF: W3C Mapping Standards

- **Relational Database to RDF**:
  - A Direct Mapping of Relational Data to RDF:
    `https://www.w3.org/TR/rdb-direct-mapping/`
  - R2RML: RDB to RDF Mapping Language: `https://www.w3.org/TR/r2rml/`
  - Each mapping involves the creation of a **SQL query** and the transformation of the results to RDF triples.

Exposing data as RDF: W3C Mapping Standards

- **Relational Database to RDF**:
  - A Direct Mapping of Relational Data to RDF: `https://www.w3.org/TR/rdb-direct-mapping/`
  - R2RML: RDB to RDF Mapping Language: `https://www.w3.org/TR/r2rml/`
  - Each mapping involves the creation of a **SQL query** and the transformation of the results to RDF triples.

- **CSV to RDF**:
  - Direct mapping from CSV to RDF (CSV2RDF): `https://www.w3.org/TR/csv2rdf/`
  - General (declarative) RDF Mapping Language (RML): `https://rml.io/specs/rml/`
  - A mapping can also be seen as a **(small) script** that creates specific RDF triples from the CSV file (*e.g.*, data frame).

Exposing data as RDF: W3C Mapping Standards

- **Relational Database to RDF**:
  - A Direct Mapping of Relational Data to RDF: `https://www.w3.org/TR/rdb-direct-mapping/`
  - R2RML: RDB to RDF Mapping Language: `https://www.w3.org/TR/r2rml/`
  - Each mapping involves the creation of a **SQL query** and the transformation of the results to RDF triples.

- **CSV to RDF**:
  - Direct mapping from CSV to RDF (CSV2RDF): `https://www.w3.org/TR/csv2rdf/`
  - General (declarative) RDF Mapping Language (RML): `https://rml.io/specs/rml/`
  - A mapping can also be seen as a **(small) script** that creates specific RDF triples from the CSV file (*e.g.*, data frame). In this module.

Exposing data as RDF: Direct Mapping Example

Automatic triples:

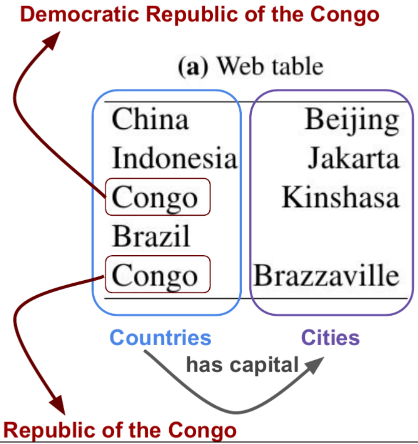
```
ex:row1 ex:col1 ''China''
ex:row1 ex:col2 ''Beijing''
ex:row2 ex:col1 ''Indonesia''
ex:row2 ex:col2 ''Jakarta''
...
```

China	Beijing
Indonesia	Jakarta
Congo	Kinshasa
Brazil	
Congo	Brazzaville

Exposing data as RDF: Enhanced Mapping/Transformation (i)

– We know the **semantics** of the data.
– **Potential automatic triples:**

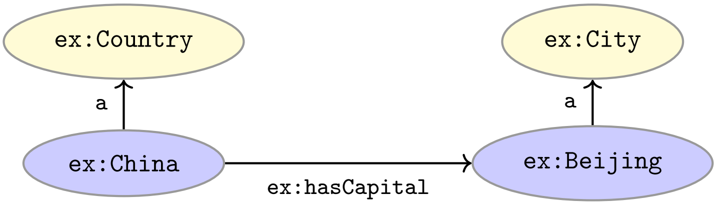
```
ex:China rdf:type ex:Country
ex:Beijing rdf:type ex:City
ex:China ex:hasCapital ex:Beijing
...
```



Exposing data as RDF: Enhanced Mapping/Transformation (ii)

Return capital of China (for $\mathcal{G}$ below):

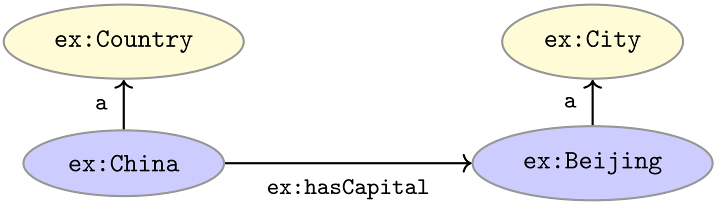
```
PREFIX ex:  <http://example.org/>
SELECT DISTINCT ?capital WHERE {
    ex:China ex:hasCapital ?capital .
}
```



Exposing data as RDF: Enhanced Mapping/Transformation (ii)

Return capital of China (for $\mathcal{G}$ below): Query Result= {ex:Beijing}

```
PREFIX ex:  <http://example.org/>
SELECT DISTINCT ?capital WHERE {
    ex:China ex:hasCapital ?capital .
}
```



Exposing data as RDF: Mappings

Mapping or Transformation $\varphi \rightsquigarrow \psi$

- φ: query over database or CSV extraction
- ψ: RDF triple template

Exposing data as RDF: Mappings

Mapping or Transformation $\varphi \rightsquigarrow \psi$

- φ: query over database or CSV extraction
- ψ: RDF triple template
- RDB to RDF mapping:

  `SELECT col1 FROM TABLE` $\rightsquigarrow$ `ex:{col1} rdf:type ex:Country`

Exposing data as RDF: Mappings

Mapping or Transformation $\varphi \rightsquigarrow \psi$

- φ: query over database or CSV extraction
- ψ: RDF triple template
- RDB to RDF mapping:
  `SELECT col1 FROM TABLE` $\rightsquigarrow$ `ex:{col1} rdf:type ex:Country`
- CSV to RDF mapping:

```
for value in data_frame[col1]:                        (φ)
    subject = "ex:" + value   #e.g., ex:China
    create_triple(subject rdf:type ex:Country)   (ψ)
```

Exposing data as RDF: Mappings

Mapping or Transformation $\varphi \rightsquigarrow \psi$

- φ: query over database or CSV extraction
- ψ: RDF triple template
- RDB to RDF mapping:

  `SELECT col1 FROM TABLE` $\rightsquigarrow$ `ex:{col1} rdf:type ex:Country`

- **CSV to RDF mapping:** (In this module)

```
for value in data_frame[col1]:                          (φ)
    subject = "ex:" + value    # e.g., ex:China
    create_triple(subject rdf:type ex:Country)          (ψ)
```

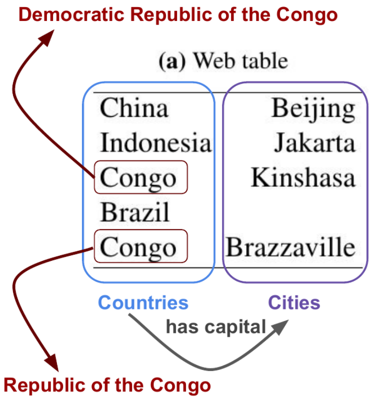
Exposing data as RDF: (Manually defined) Mappings

– Column 1 is composed by Countries.

```
for value in data_frame[col1]:
    subject = "ex:"  + value   #e.g., ex:China
    create_triple(subject rdf:type ex:Country)
```

– Column 2 entities are the capitals of column 1 entities.

```
for row in data_frame:
    subj = "ex:"  + row[col1]   #e.g., ex:China
    obj = "ex:"  + row[col2]   #e.g., ex:Beijing
    create_triple(subj ex:hasCapital obj)
```



Semantic Understanding of Tabular Data:
Automating the Process

Adding Semantics to Tabular Data: Basic Tasks

- Matching a cell to a KG entity (**CEA task** - Cell-Entity Annotation)
- Assigning a semantic type (*e.g.*, a KG class) to an (entity) column (**CTA task** - Column-Type Annotation)
- Assigning a KG property to the relationship between two columns (**CPA task** - Columns-Property Annotation)

Ernesto Jiménez-Ruiz and others. **SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems**. ESWC 2020

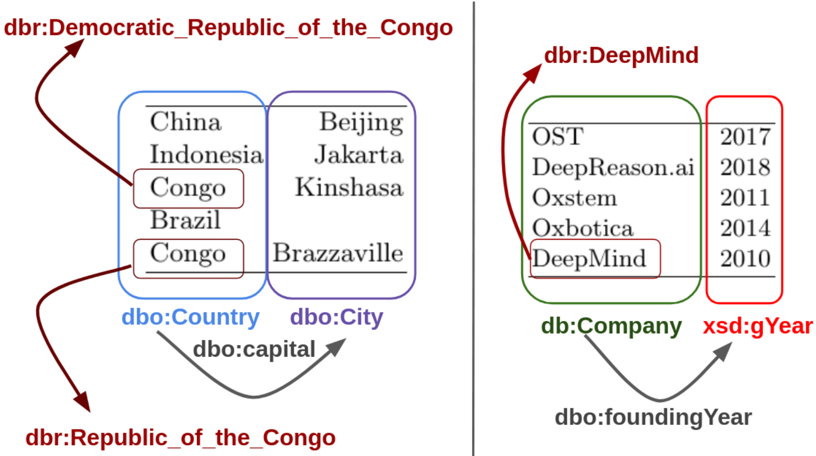
Adding Semantics to Tabular Data: Basic Tasks

- Matching a cell to a KG entity (**CEA task** - Cell-Entity Annotation)
- Assigning a semantic type (*e.g.*, a KG class) to an (entity) column (**CTA task** - Column-Type Annotation)
- Assigning a KG property to the relationship between two columns (**CPA task** - Columns-Property Annotation)

† *For a semi-automatic process, we assume the existence of a (possibly incomplete)* ***Knowledge Graph (KG)*** *relevant to the domain.*
‡ *When transforming to RDF, if no KG matching then create a fresh entity URI.*

Ernesto Jiménez-Ruiz and others. **SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems**. ESWC 2020

Adding Semantics to Tabular Data: Basic Tasks (with DBpedia)



SemTab Challenge

- **Systematic evaluation** of Tabular Data to KG matching systems.
- Evaluates the **three basic tasks**: CTA, CEA and CPA.
- Relies on:
  - an **automatic** dataset generator, and
  - **manually** curated datasets.
- **Target KGs**: DBpedia, Wikidata and Schema.org.
- Co-organised and sponsored by **IBM Research**.

SemTab: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching:
`http://www.cs.ox.ac.uk/isg/challenges/sem-tab/`

Semantic Understanding of Tabular Data Techniques

Common Techniques

- **Pre-processing**: spelling error, stopwords, unicode fixing, etc.
- **Regular expressions** to identify data formats (*e.g.*, numbers, phones, dates, names).
- **NLP techniques** to process textually-rich fields.

Common Techniques

- **Pre-processing**: spelling error, stopwords, unicode fixing, etc.
- **Regular expressions** to identify data formats (*e.g.*, numbers, phones, dates, names).
- **NLP techniques** to process textually-rich fields.
- **Fuzzy search** over a KG
  - Via online look-up services
  - Or local indexes
- Access to the **KG's SPARQL Endpoint** (local or online)

Common Techniques

- **Pre-processing**: spelling error, stopwords, unicode fixing, etc.
- **Regular expressions** to identify data formats (*e.g.*, numbers, phones, dates, names).
- **NLP techniques** to process textually-rich fields.
- **Fuzzy search** over a KG
  - Via online look-up services
  - Or local indexes
- Access to the **KG's SPARQL Endpoint** (local or online)
- **Lexical similarity** (*e.g.*, Levenshtein)
- Word and KG **embeddings**

Common Knowledge Graphs

Wikidata: `https://www.wikidata.org/`

- $>$90 million entities
- Free and public (anyone can edit)

DBpedia: `https://dbpedia.org/`

- $>$900 million triples
- Extracted from Wikipedia

Google KG: `https://developers.google.com/knowledge-graph`

- Private, only accessible via look-up
- $>$1,000 million entities

Fuzzy Search: KG look-up Services

- Given a string (*e.g.*, “Congo”)
- Return a set of candidate KG entities, *e.g.*,
  `http://dbpedia.org/resource/Republic_of_the_Congo`
  `http://dbpedia.org/resource/Congo_River`
- Typical starting point for CEA and CTA tasks
- DBPedia, Wikidata and Google KG provide look-up services via a REST API.
- Some systems have built their own local index for fuzzy search.

GitHub repositories: `https://github.com/city-knowledge-graphs`

Lexical Processing and Similarity

- **Datatype prediction**, *e.g.*, ptype:
  `https://github.com/alan-turing-institute/ptype`
- **Spelling corrector:** `https://norvig.com/spell-correct.html`

Lexical Processing and Similarity

- **Datatype prediction**, *e.g.*, ptype:
  `https://github.com/alan-turing-institute/ptype`
- **Spelling corrector:** `https://norvig.com/spell-correct.html`
- **Lexical similarity:**
  - Levenshtein distance:
    `levenshtein('Congo', 'Republic of Congo')=12`
  - Jaro Winkler:
    `jaro_winkle('Congo', 'Republic of Congo')=0.0`
    `jaro_winkle('Congo', 'Congo Republic')=0.893`
  - **I-Sub**:
    `isub('Congo', 'Republic of Congo')=0.727`

Access to KG SPARQL Endpoint

- Get additional **contextual information**:
  - Additional type information (*e.g.*, `dbr:London rdf:type dbo:City`)
  - Relationships (*e.g.*, `dbr:London dbo:country dbr:United_Kingdom`)
  - Members of a type
- Access via **SPARQL queries** (no fuzzy search)
- Typically required for:
  - the **CPA task**
  - **disambiguation** in CTA and CEA tasks

GitHub repositories: `https://github.com/city-knowledge-graphs`

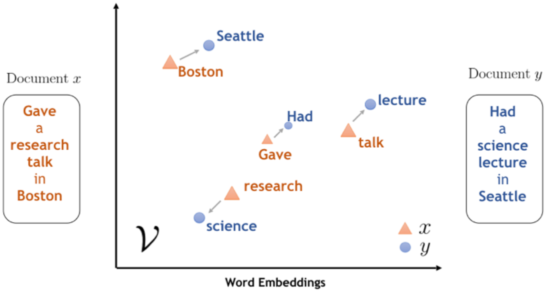
Word and KG Embeddings: Capturing Context

- **Embeddings**: representation in the form of a real-valued vector.
- Very useful to **capture the meaning/semantics** of a word (or a KG entity).
- Comparison among vectors via **Cosine similarity** (*e.g.*, between vectors for 'Congo' and 'Republic of Congo')

Word and KG Embeddings: Capturing Context

- **Embeddings**: representation in the form of a real-valued vector.
- Very useful to **capture the meaning/semantics** of a word (or a KG entity).
- Comparison among vectors via **Cosine similarity** (*e.g.*, between vectors for 'Congo' and 'Republic of Congo')
- **Precomputed word embeddings:**
  - `https://wikipedia2vec.github.io/wikipedia2vec/pretrained/`
  - `https://fasttext.cc/docs/en/pretrained-vectors.html`
  - `https://www.analyticsvidhya.com/blog/2020/03/pretrained-word-embeddings-nlp/`
- **Precomputed KG embeddings:**
  - Wikidata: `http://139.129.163.161/index/toolkits#pretrained-embeddings`
  - DBpedia: `http://www.kgvec2go.org/`

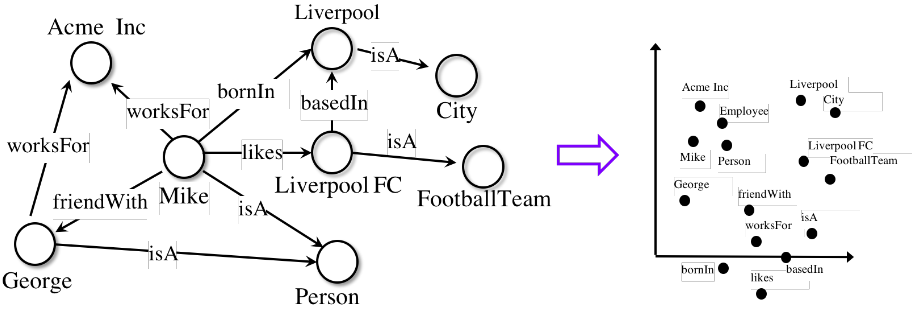
Word Embeddings: Example



Systems like `Word2Vec` require a corpus of documents as training.

Example from: `https://dsgiitr.com/blogs/word_embeddings/`

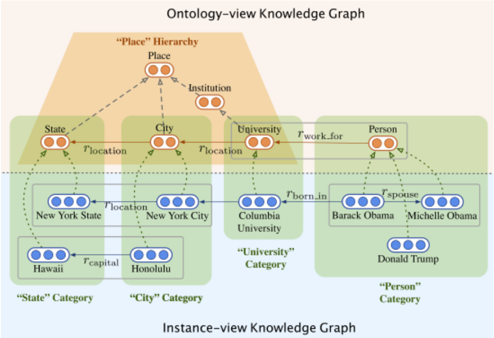
Knowledge Graph Embeddings: Example (i)

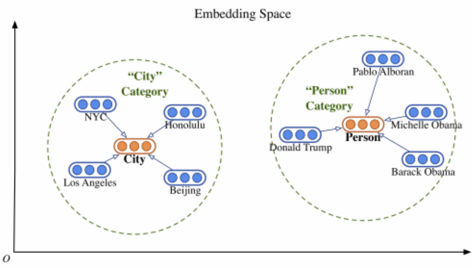


KG Embedding Systems exploit the neighbourhood of an entity to calculate its vector.

Example from: `https://docs.ampligraph.org/en/1.0.3/`

Knowledge Graph Embeddings: Example (ii)





KG Embedding Systems exploit the neighbourhood of an entity to calculate its vector.

Example from: Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. KDD 2019.

Semantic Understanding of Tabular Data:
User Interfaces

OpenRefine

- `https://openrefine.org/`
- Previously known as *Google Refine*.
- **Interface to support** the cleaning and transformation of messy data.
- Includes a **reconciliation service** to link the data with a KG (*e.g.*, Wikidata is the default installation).
- In this module we will not use OpenRefine, but perform our own reconciliation programmatically.

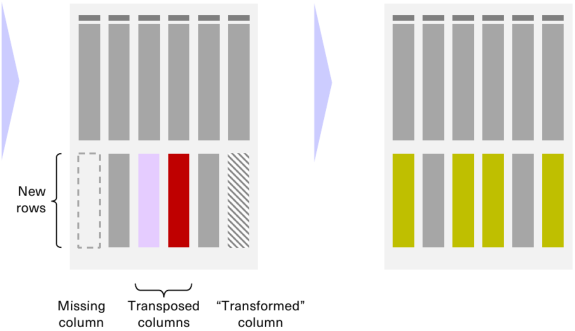
Applications

Examples of Applications of KGs and Semantics

- Data Wrangling (Alan Turing Institute)
- Data Access in Oil & Gas Industry
- Data Access and Prediction in Ecotoxicology
- Semantic Reasoning and Ontologies for Autonomous Vehicles
- FAIR Implementation for Life Science
- Amazon Product Graph

AIDA project: Data Wrangling with DataDiff

- – The structure of a dataset may change after an update
- – Changes may break the analytical pipeline.
- ✓ Datadiff **identifies and patches** these changes.
- ✗ Limitation: **exhaustive comparison** of columns.
- ✓ Semantic table understanding **may limit the comparison**.



Data Diff: Interpretable, Executable Summaries of Changes in Distributions for Data Wrangling. C. Sutton, T. Hobson, J. Geddes and R. Caruana. In KDD 2018.

Data Access in Oil & Gas Industry

- Data access currently takes **30-70%** of the engineers' time.
- Data cannot be moved from the original sources.
- The EU project Optique advocated for an **Ontology-Based Data Access** (OBDA) process. Requirements:
  - Domain ontology.
  - Mappings to create a virtual KG.

Ontology Based Data Access in Statoil. Journal of Web Semantics, 44, pp. 3-36
`https://openaccess.city.ac.uk/id/eprint/22959/`

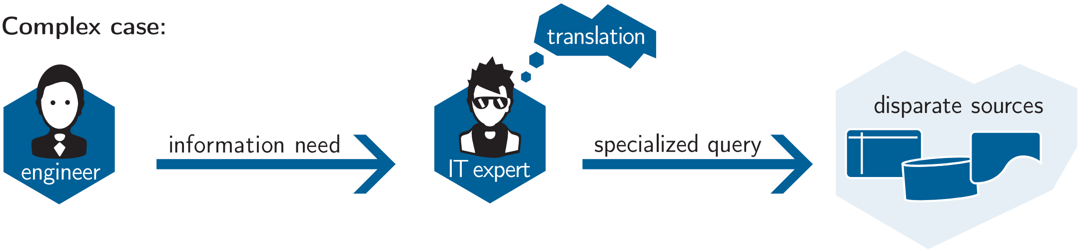
Data Access in Oil & Gas Industry: Limitations

Simple case:

Problem when the information needs fall outside predefined-queries

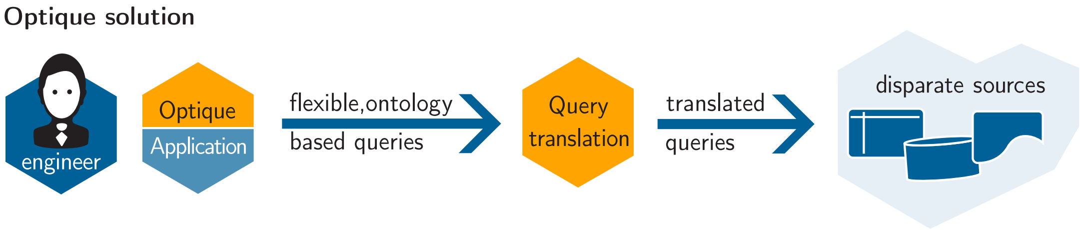
Data Access in Oil & Gas Industry: Limitations

Complex case:



The process may take several days

Data Access in Oil & Gas Industry: Optique Solution

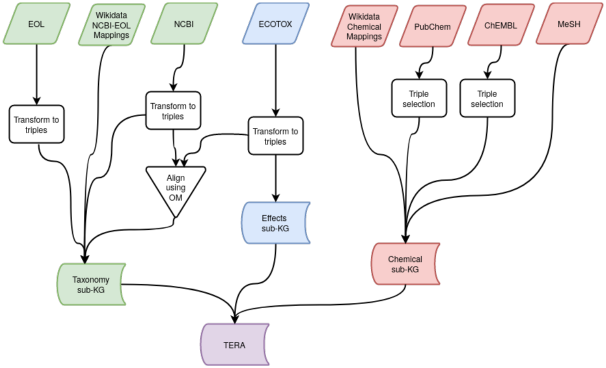


Optique Solution

1. Mediator to create ontology-driven queries (SPARQL).
2. Mediator to translate SPARQL queries into SQL queries.
3. Effort required to create the ontology and maintain the mappings (modular approach).

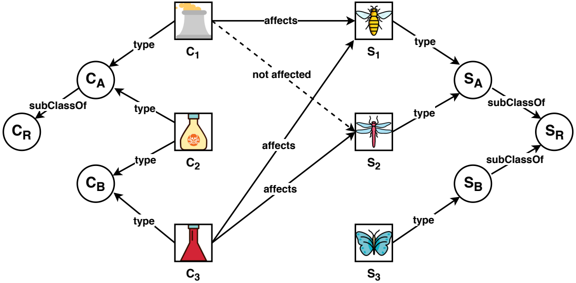
TERA: A KG for Ecotoxicology. Integration and Data Access.

- **Integrates** disparate sources about species, chemicals and effect data.
- Enhances **data access**.



TERA: A KG for Ecotoxicology. Prediction.

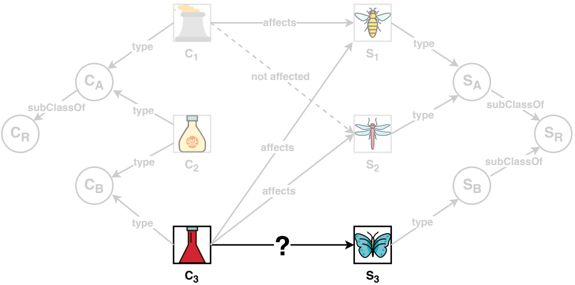
– Drives the **prediction of adverse biological effects** of chemicals via KG embeddings.



Resources and publications: `https://github.com/NIVA-Knowledge-Graph/`

TERA: A KG for Ecotoxicology. Prediction.

– Drives the **prediction of adverse biological effects** of chemicals via KG embeddings.



Resources and publications: `https://github.com/NIVA-Knowledge-Graph/`

Semantic Reasoning and Ontologies for Autonomous Vehicles

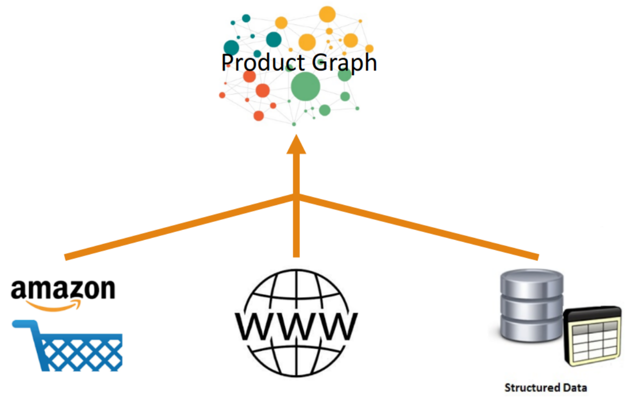
Oxford Semantic Technologies: `https://www.oxfordsemantic.tech/blog/reasonable-vehicles-rule-the-road`

FAIR Implementation for Life Science

Knowledge Graphs play a key role for annotating data.

Pistoia Alliance FAIR Toolkit: `https://fairtoolkit.pistoiaalliance.org/`

Amazon Product (Knowledge) Graph



Challenges and Innovations in Building a Product Knowledge Graph: `http://lunadong.com/talks/PG.pdf`
`https://www.amazon.science/blog/building-product-graphs-automatically`

Laboratory: From CSV to a KG

Support Codes

- `https://github.com/city-knowledge-graphs`
- Lookup
- SPARQL Endpoint
- Lexical similarity
- CSV management

Wonder.me

- Use of **wonder.me**:
  - `https://app.wonder.me/?spaceId=d39fd38d-5cc3-48fd-9952-5990e851b681`
  - Short URL: `https://bit.ly/in3067-inm713-online-lab`