MONITORING RESEARCH AND INNOVATION FROM HETEROGENEOUS SOURCES USING KNOWLEDGE GRAPHS

Vanni Zavarella
University of Cagliari - Department of Mathematics and Computer Science

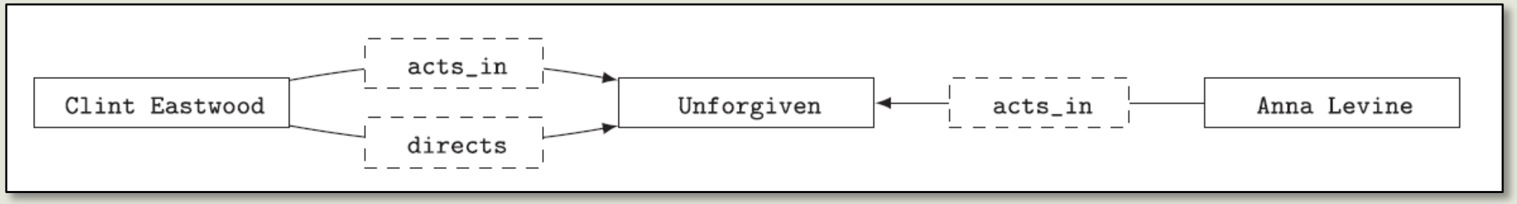
DATAI – UNAV
November 22nd, 2023

Outline

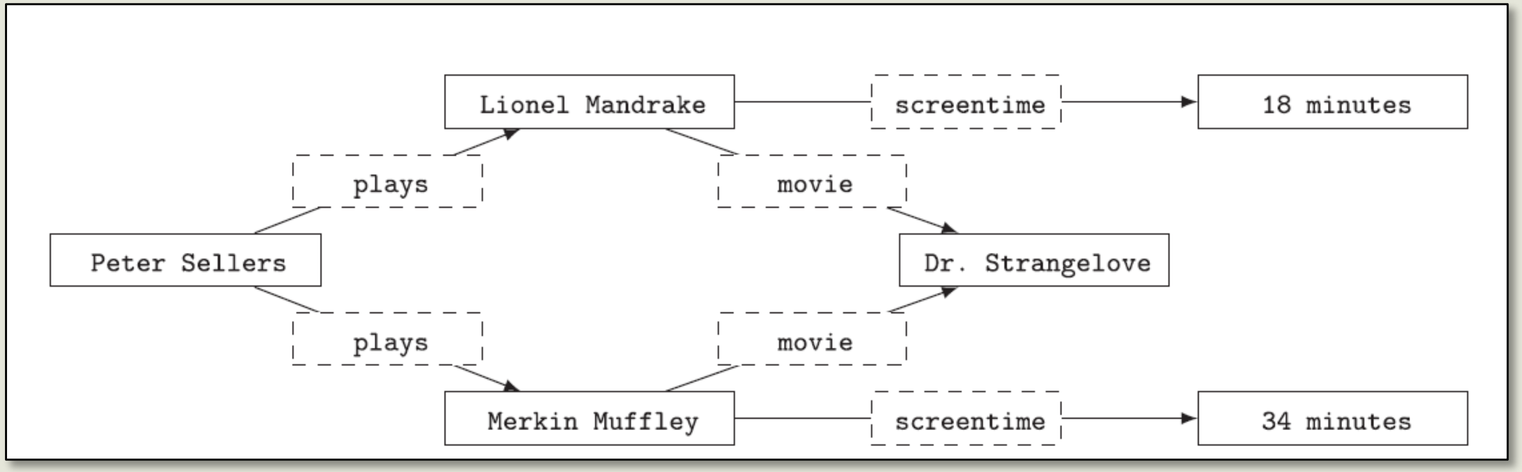
- Knowledge Graphs
- Use Case: Scholarly Domain
- Pipeline
- Data Collection
- Data Preprocessing
- Entity Extraction and Linking
- Relation Extraction and Clustering
- Triple Store and Data storing
- Current Limitations
- Ongoing developments

Knowledge Graphs: Definition

- Directed edge-labelled graphs representation of a target domain
- Formally, a tuple: $G \coloneqq (V, E, L)$ with
  - V a finite set of nodes
  - L a finite set of labels
  - $E \subseteq V \times L \times V$ is a set of edges
- More flexible than tabular data representation
- No topology specifications



- $V = \{Clint\_Eastwood, Anna\_Levin, Unforgiven\}$
- $L = \{acts\_in, directs\}$
- *E = {(Clint_Eastwood,acts_in,Unforgiven), {(Clint_Eastwood,directs,Unforgiven), {(Anna_Levine,acts_in,Unforgiven)}*



Knowledge Graphs: Property Graphs

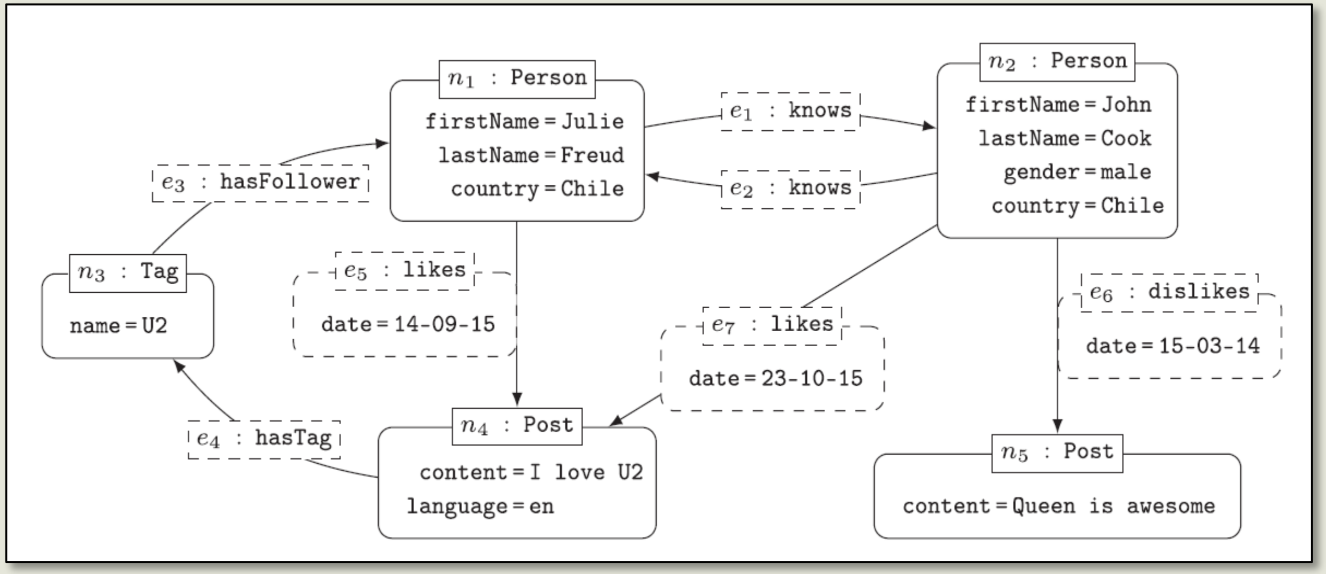
- Property graphs: both nodes and edges can be labelled, associated with unique identifiers and optionally with a set of attribute/value pairs

A tuple: $G \coloneqq (V, E, L, Prop, Val, \rho, \lambda, \sigma)$

V a finite set of nodes, E a finite set of edges with $V \cap E = \emptyset$

$\rho: E \rightarrow (V \times V)$ and $\lambda: (V \cup E) \rightarrow L$ are total functions

$\sigma: (V \cup E) \times Prop \rightarrow Val$ is a partial function with Prop,Val finite sets of properties and values



$V = \{n_1, \ldots n_5\} \quad E = \{e_1, \ldots e_7\}$
$\rho(e_1) = (n_1, n_2), \ldots, \rho(e_7) = (n_2, n_4)$

$\lambda(n_1) = Person, \ldots, \lambda(n_5) = Post, \lambda(e_1) = knows, \ldots, \lambda(e_7) = likes$

$\sigma(n_1, firstName) = \text{Julie}$, $\sigma(n_1, lastName) = \text{Freud}$, $\sigma(n_5, content) =$ Queen is awesome, $\sigma(e_5, date) = 14.09.15$

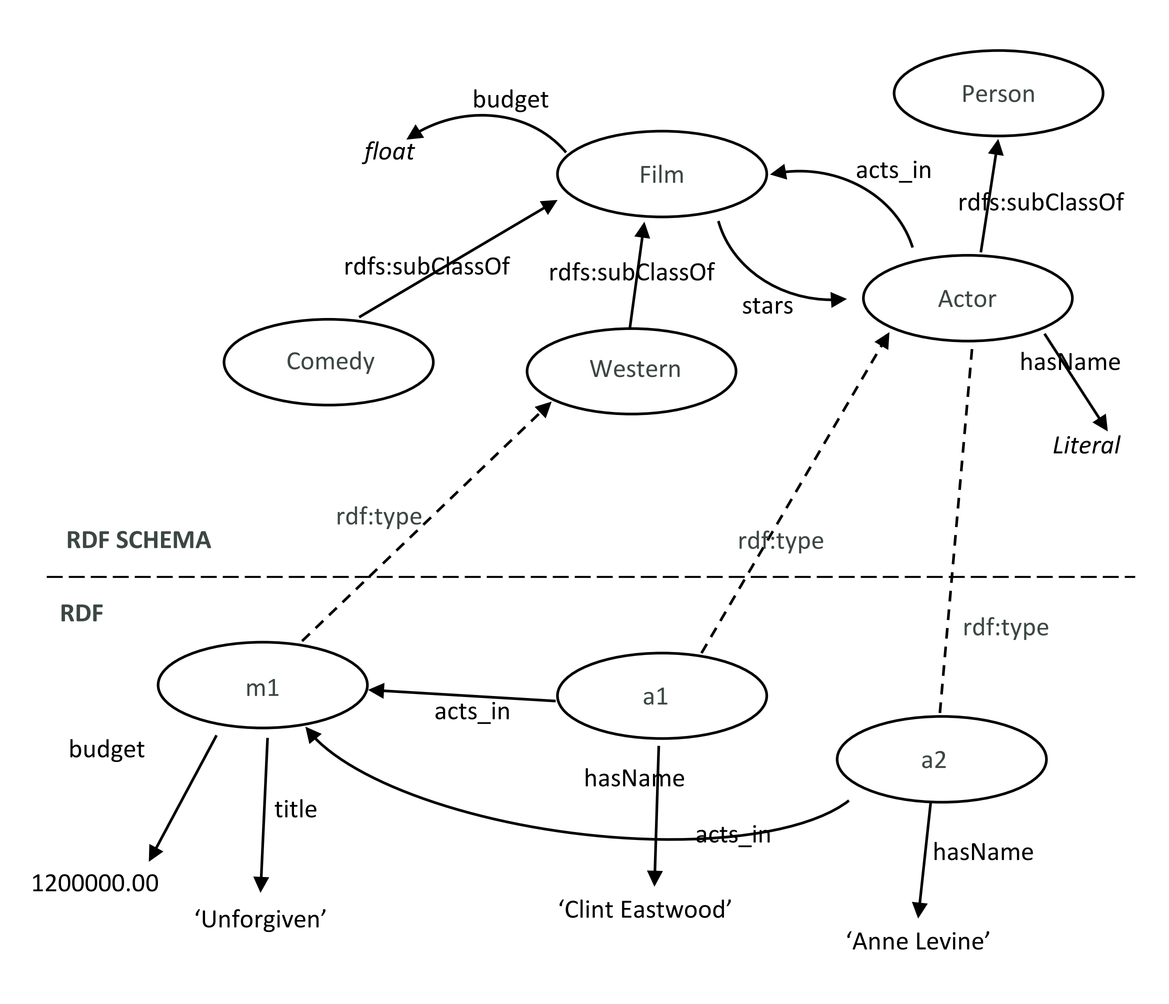
Knowledge Graphs: RDF and RDF Schema

- KGs interoperability requires imposing a semantics of the nodes/relation labels
- Different languages allow defining axioms of various complexity
- A minimal formalisation for DEL graphs is RDF/RDF Schema
- RDF a standardized data model for DEL graphs with restrictions on node/edge identifiers:
  - Nodes can be Uniform Resource Identifiers, XML Schema Datatypes(Literals, Date, Integer, etc.) or blank nodes
  - HTTP URIs for nodes and edges can be looked up by web-servers to return RDF descriptions (Semantic Web principle)
  - URIs are organized in namespaces (prefixed)
  - rdf:type, rdf:Property
- RDFS a metalanguage for defining the semantics of the terms in a RDF KG (an Ontology)
  - rdfs:Resource, rdfs:subClassOf, rdfs:Class, rdfs:Domain, rdfs:Range, rdfs:subPropertyOf, rdf:Statement, etc.

Knowledge Graphs: RDF example

```
@prefix mdb-ont : <http://movie-database/ontology#> .
@prefix mdb :  <http://movie-database/resource> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

xsd:date rdf:type rdfs:Datatype .

mdb-ont:Actor rdf:type owl:Class ;
     rdfs:subClassOf :Performer ,
              <http://xmlns.com/foaf/0.1/Person>.

mdb-ont :Film rdf:type owl:Class ;
     rdfs:label “Film" .

mdb-ont :Western rdf:type owl:Class ;
    rdfs:subClassOf :Film.

mdb-ont :Award rdf:type owl:Class.

mdb-ont:acts_in rdf:type owl:ObjectProperty ;
       rdfs:subPropertyOf :perfomsIn ;
       rdfs:domain mdb-ont :Actor ;
       rdfs:range mdb-ont :Film;
       owl:minCardinality 1 .
```

```
mdb-ont:stars rdf:type owl:ObjectProperty ;
      rdfs:domain:Film ;
      rdfs:range mdb-ont:Actor ;
      owl:inverseOf  mdb-ont:acts .

mdb-ont:budget rdf:type owl:DatatypeProperty ;
     rdfs:domain mdb-ont :Film ;
     rdfs:range xsd:float .

mdb-ont:title rdf:type owl:DatatypeProperty ;
     rdfs:domain mdb-ont :Film ;
     rdfs:range rdfs:Literal.

mdb-ont:hasName rdf:type owl:DatatypeProperty ;
     rdfs:domain mdb-ont :Film ;
     rdfs:range rdfs:Literal.

mdb:m1  rdf:type owl:NamedIndividual ,
          mdb-ont :Film ;
          mdb-ont:title   ‘Unforgiven’.

mdb:a1  rdf:type owl:NamedIndividual ,
          mdb-ont :Actor ;
          mdb-ont:hasName  ‘Clint Eastwood’.

mdb:a2  rdf:type owl:NamedIndividual ,
          mdb-ont :Film ;
          mdb-ont:hasName  ‘Anne Levine ’.
```



Querying Graphs: Graph Patterns

Query*:* Find all co-stars of a movie in graph G

A graph pattern is a tuple Q=(V,E,L) where V,L ⊆ Term

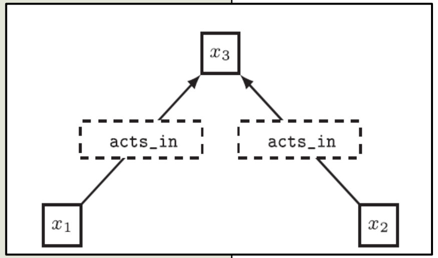
$\mathrm{Var} \cap \mathrm{Const} = \emptyset$, $\mathrm{Term} = \mathrm{Const} \cup \mathrm{Var}$

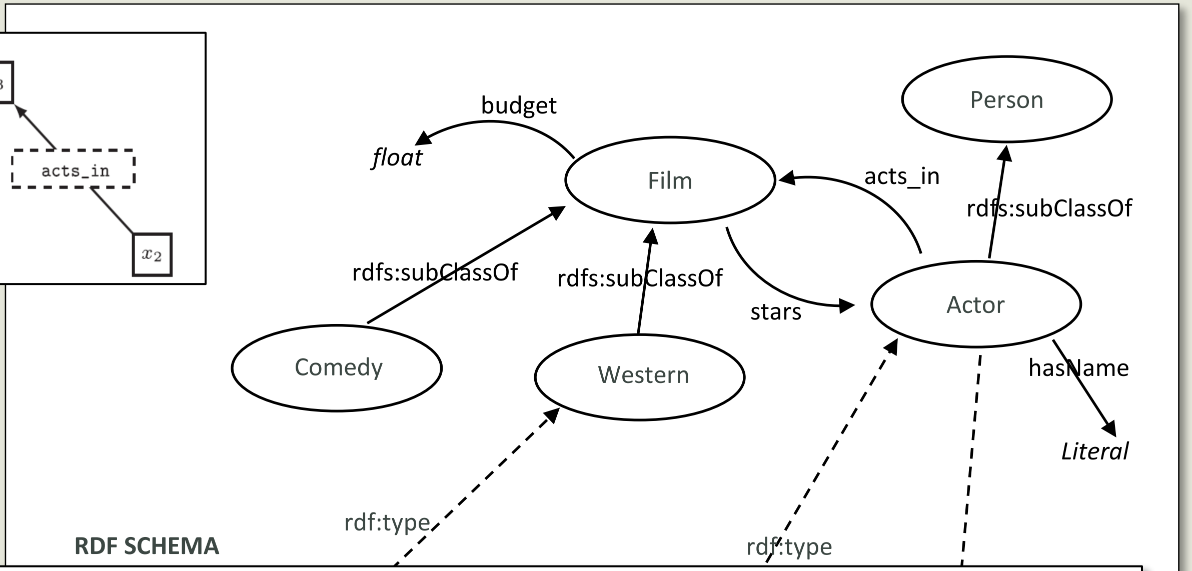
$E \subseteq V \times L \times V$ is a set of triples

Var(Q) = all variables in Q

The evaluation of Q over the data graph G:
$Q(G) = \{\mu | \mu(Q) \subseteq G, \mathrm{dom}(\mu) = \mathrm{Var}(Q)\}$

x_1	x_2	x_3
Clint Eastwood	Anna Levine	Unforgiven
Anna Levine	Clint Eastwood	Unforgiven
Clint Eastwood	Clint Eastwood	Unforgiven
Anna Levine	Anna Levine	Unforgiven





SPARQL

```
PREFIX : <http://ex.org/#>
SELECT ?x1 ?x2
WHERE {
        ?x1 :acts_in ?x3 . ?x1 :type :Person .
        ?x2 :acts_in ?x3 . ?x2 :type :Person .
        ?x3 :type :Film .}
FILTER(?x1 != ?x2)
```

Querying Graphs: Path Queries

Query: find the posts that are liked by friends of friends of Julie and have a tag that Julie follows.

Path expression: $l \in L$

if r_1, r_2 are PE, $r_1^-, r_1^*, r_1 \cdot r_2$, $r_1 | r_2$

Path expression evaluation

Given $G = (V, E, L)$ and PE r , evaluation r[G]:

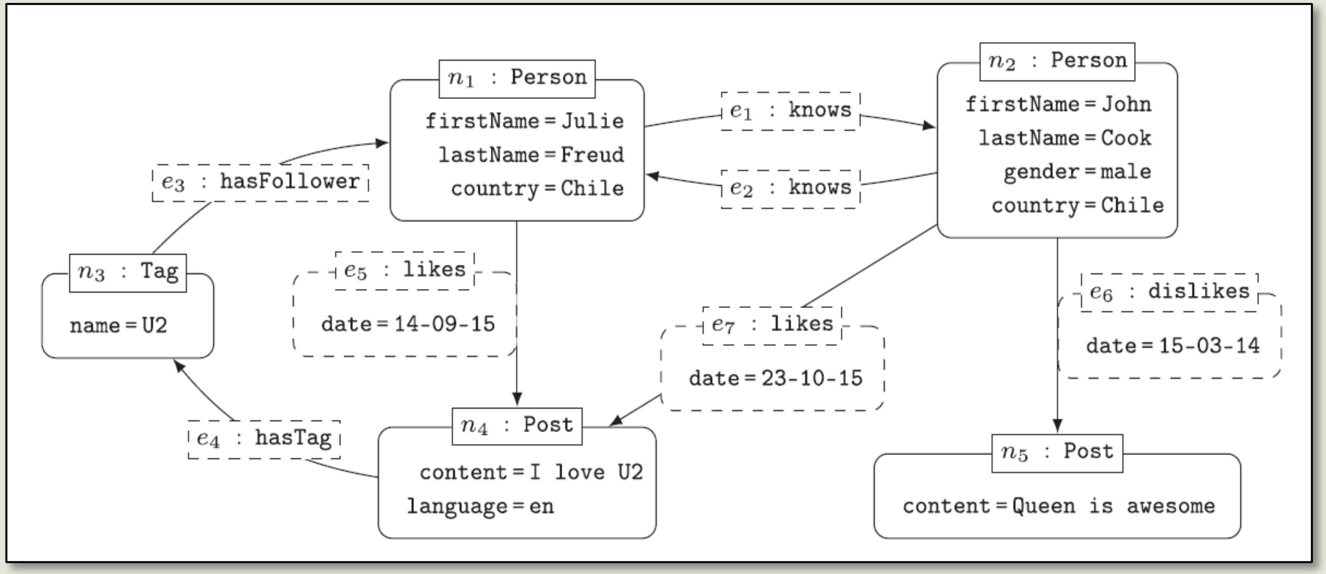
$r[G] \coloneqq \{(u, v) \mid (u, r, v) \in E\}$ (for $r \in$ Con)

$r^- [G] \coloneqq \{(u, v) \mid (v,u) \in r[G]\}$

$r_1 \mid r_2 [G] \coloneqq r_1[G] \cup r_2[G]$

$r_1 \cdot r_2[G] \coloneqq \{(u, v) \mid \exists w \in V : (u,w) \in r_1 [G]$ and $(w, v) \in r2 [G]\}$

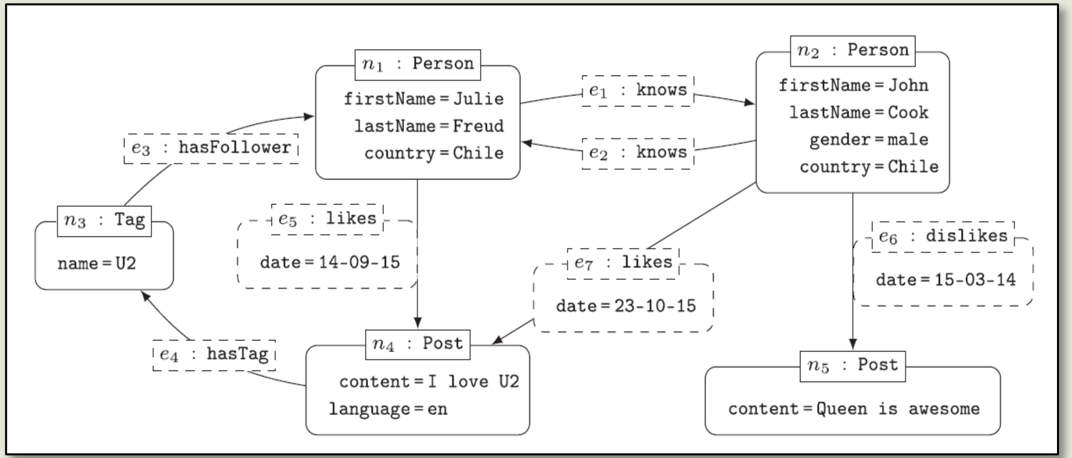
$r^*[G] \coloneqq \bigcup_{n\in\mathbb{N}^+} r^n[G]$ with r_n the n^{th}-concatenation of r

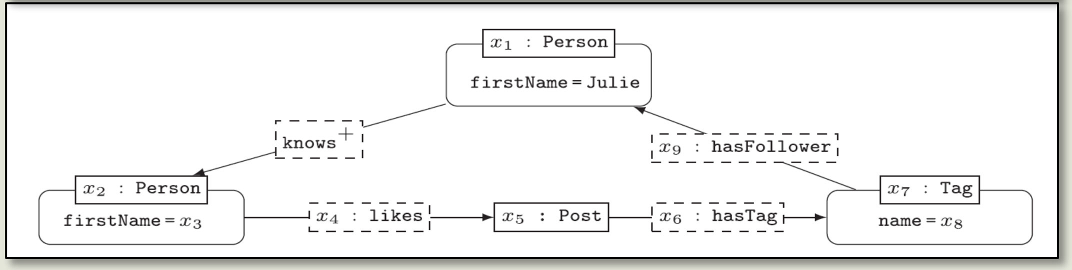


Querying Graphs: Navigational Graph Patterns

Query: find the posts that are liked by friends of friends of Julie and have a tag that Julie follows.

- Graph patterns can be combined with regular path queries to create complex query language





SPARQL

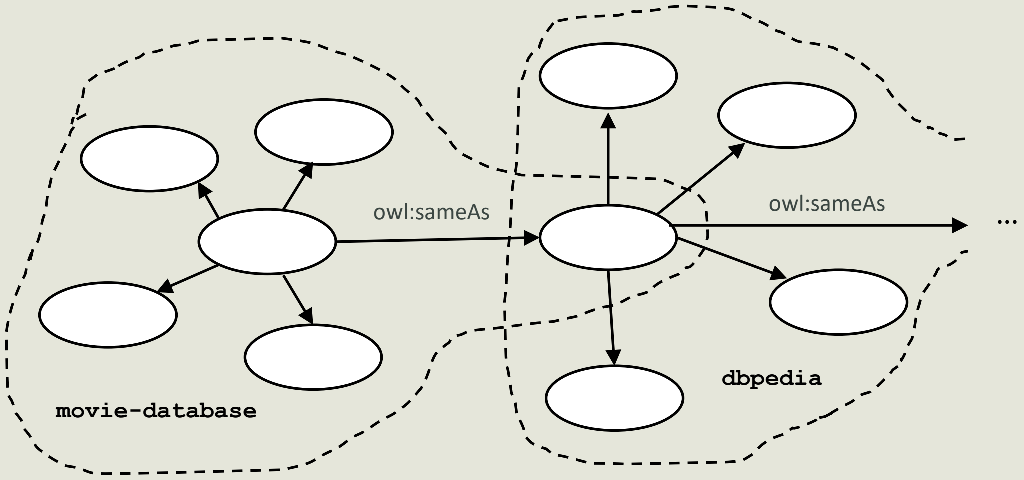
```
SELECT ?x5
WHERE {
        :Julie :knows+ ?x2 . ?x2 :type :Person .
        ?x2 :likes ?x5 . ?x5 :type ?Post .
        ?x5 :hasTag ?x7 . ?x7 :type ?Tag.
        ?x7 :hasFollower :Julie.
        }
```

Bridging Knowledge Graphs

- We can reference distinct KGs in the namespace declarations
- we use the OWL predicate owl:sameAs to state equality of individuals from different ontologies

  `mdb:Unforgiven owl:sameAs dbpedia:Unforgiven`

- SPARQL queries are evaluated wrt a RDF dataset
  - 1 default graph
  - a set of named graphs
- Now we can access different information about the same individuals as encoded in different KGs




```
SELECT ?x1 ?earnings
FROM <http://movie-database>
FROM NAMED <http:dbpedia.org>
WHERE { GRAPH <http:dbpedia.org>
          {?x1 owl:sameAs ?x2 .
           ?x2 :earned ?earnings .
                    }
          ?x1 :hasName ‘Unforgiven’. ?x1 :type :Film
       }
```

Knowledge Graphs: Reification

- By default a triple represents a fact that holds True in domain
- No way to distinguish the fact from the assertion about that fact and the related properties
- Turn a predicate edge into a node of type rdf:Statement and add 3 native triples for subj,pred, obj
- It allow add attributes from different meta-ontologies describing the Provenance of information (PROV), TIME, etc.

```
:Unforgiven :stars :Clint_Eastwood
```

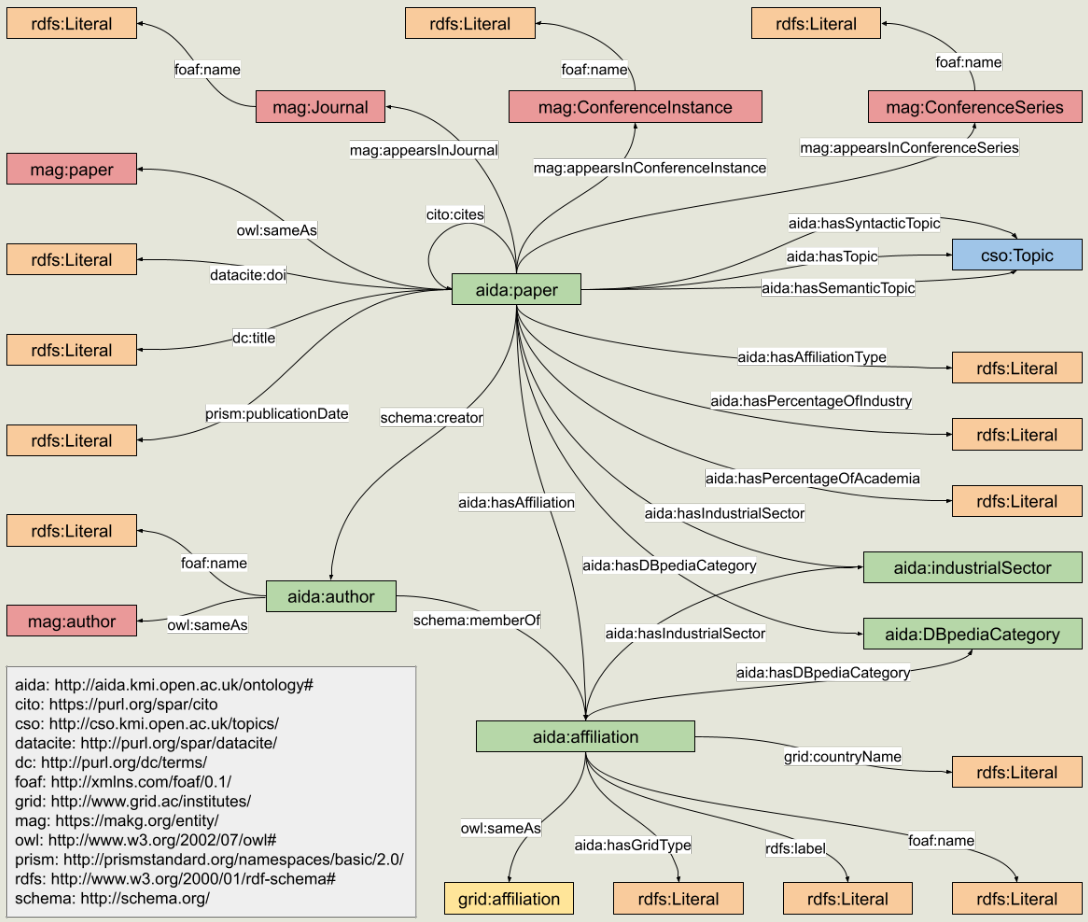
```
:statement_01 a rdf:Statement ;
              rdf:subject :Unforgiven ;
              rdf:predicate :stars ;
              rdf:object : Clint_Eastwood ;
              time:validFrom 1976 ;
              prov:wasDerivedFrom
              …
             .
```

Scholarly Domain KGs

- Document-centric Knowledge Graphs: based on metadata like authors, titles, organizations, citations, controlled-vocabulary topic terms
  - Microsoft Academic Graph
  - Semantic Scholar
  - OpenAlex
  - AIDA
- Content-based Knowledge Graphs: knowledge triples extracted from Abstract/Full Text
  - Open Research Knowledge Graph
  - Computer Science Knowledge Graph

Scholarly Domain KGs

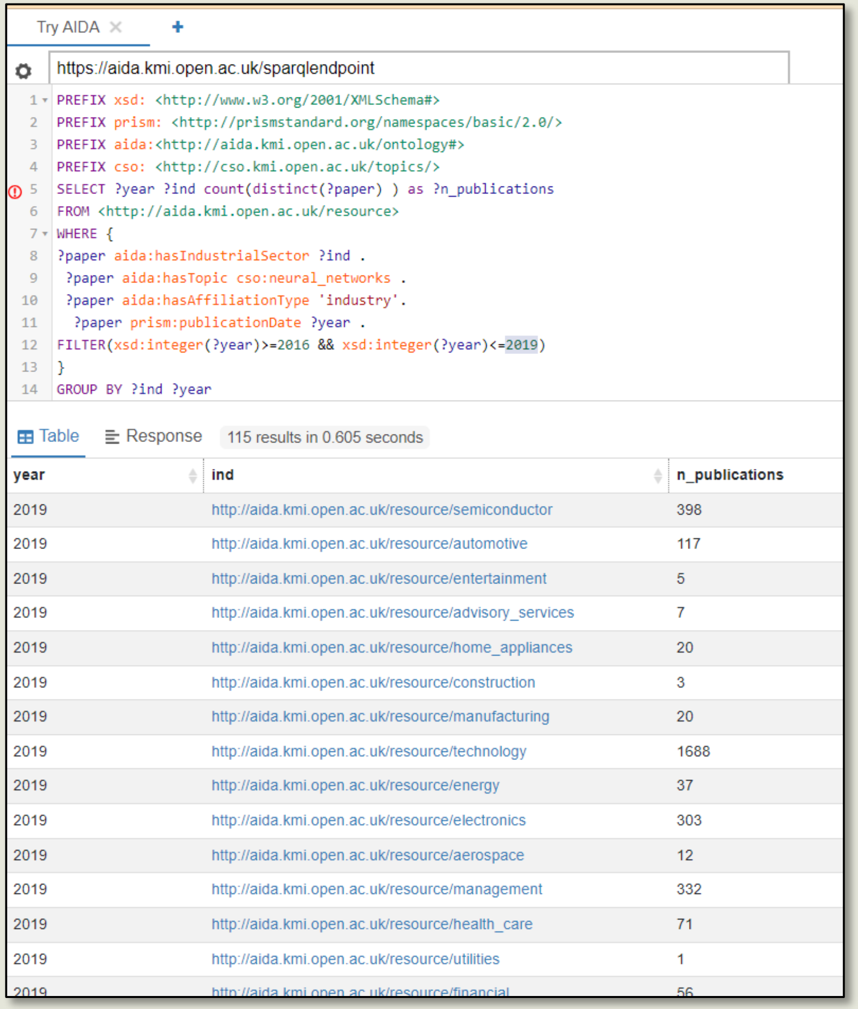
- Academia/Industry DynAmics (AIDA) Knowledge Graph: 21M publications and 8M patents in Computer Science
- Main classes: paper/patent,cso:Topic,author,affiliation,affiliationType(academia,industry,collaborative),industrialSector
- Main relations: hasTopic, hasIndustrialSector, hasAffiliation, hasAffiliationType,schema:creator,schema:memberOf
- Uses a very granular topic tagger for CS (14k topics)



Scholarly Domain KGs

- *it supports extracting analytical data about the relation between Research and Industry*

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prism: <http://prismstandard.org/namespaces/basic/2.0/>
PREFIX aida:<http://aida.kmi.open.ac.uk/ontology#>
PREFIX cso: <http://cso.kmi.open.ac.uk/topics/>

SELECT ?year ?ind count(distinct(?paper) ) as ?n_publications
FROM <http://aida.kmi.open.ac.uk/resource>
WHERE {
?paper aida:hasIndustrialSector ?ind .
 ?paper aida:hasTopic cso:neural_networks .
 ?paper aida:hasAffiliationType 'industry'.
  ?paper prism:publicationDate ?year .
FILTER(xsd:integer(?year)>=2016 && xsd:integer(?year)<=2019)
}
GROUP BY ?ind ?year
```



year	ind	n_publications
2019	http://aida.kmi.open.ac.uk/resource/semiconductor	398
2019	http://aida.kmi.open.ac.uk/resource/automotive	117
2019	http://aida.kmi.open.ac.uk/resource/entertainment	5
2019	http://aida.kmi.open.ac.uk/resource/advisory_services	7
2019	http://aida.kmi.open.ac.uk/resource/home_appliances	20
2019	http://aida.kmi.open.ac.uk/resource/construction	3
2019	http://aida.kmi.open.ac.uk/resource/manufacturing	20
2019	http://aida.kmi.open.ac.uk/resource/technology	1688
2019	http://aida.kmi.open.ac.uk/resource/energy	37
2019	http://aida.kmi.open.ac.uk/resource/electronics	303
2019	http://aida.kmi.open.ac.uk/resource/aerospace	12
2019	http://aida.kmi.open.ac.uk/resource/management	332
2019	http://aida.kmi.open.ac.uk/resource/health_care	71
2019	http://aida.kmi.open.ac.uk/resource/utilities	1
2019	http://aida.kmi.open.ac.uk/resource/financial	56

Scholarly Domain KGs

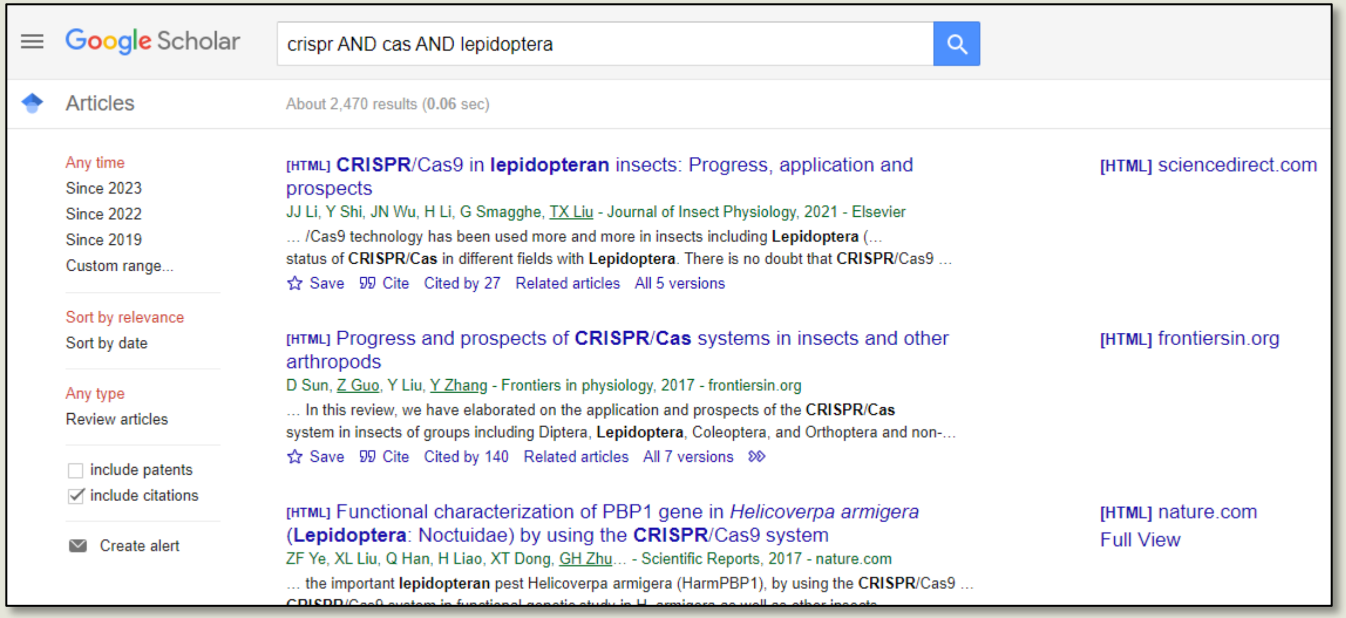
Use case: answering research questions on research entities

Entity: CRISPR/Cas9 method

Question: Precision/Safety

Constraint: on butterflies

- Overwhelmed by result size
- Recall depends on query term choice



Scholarly Domain KGs

- Need of an explicit representation of research knowledge in the papers, aside of topic labels in domain vocabulary
- Support for semantic queries such as:
  - which **metrics** are **used** to **evaluate dimensionality reduction**?
  - which **benchmarks** are **used** for **fake news detection**?
  - …
- This requires automatic detection of Research entities and relations:
  - ont:Task (genome editing, nonlinear dimensionality reduction, fake news detection)
  - ont:Method (CRISPR/Cas9, UMAP,…)
  - ont:Dataset (e.g. LIAR)
  - ont:UsedFor, ont:EvaluateOn

Scholarly Domain KGs

- **Problem Statement:**

  given a document collection $D = \{d_1, \dots d_n\}$, build a model :

  $\gamma: D \rightarrow G$

  with $G \coloneqq (E, T, R)$

  E a finite set of nodes (domain-specific research entities)

  R a finite set of relation labels (domain-specific research relations)

  $T \subseteq E \times R \times E \times \mathbb{P}(D)$ is the set of triples of the form $< e_i, r, e_j >$ referencing the subsets of documents generating them

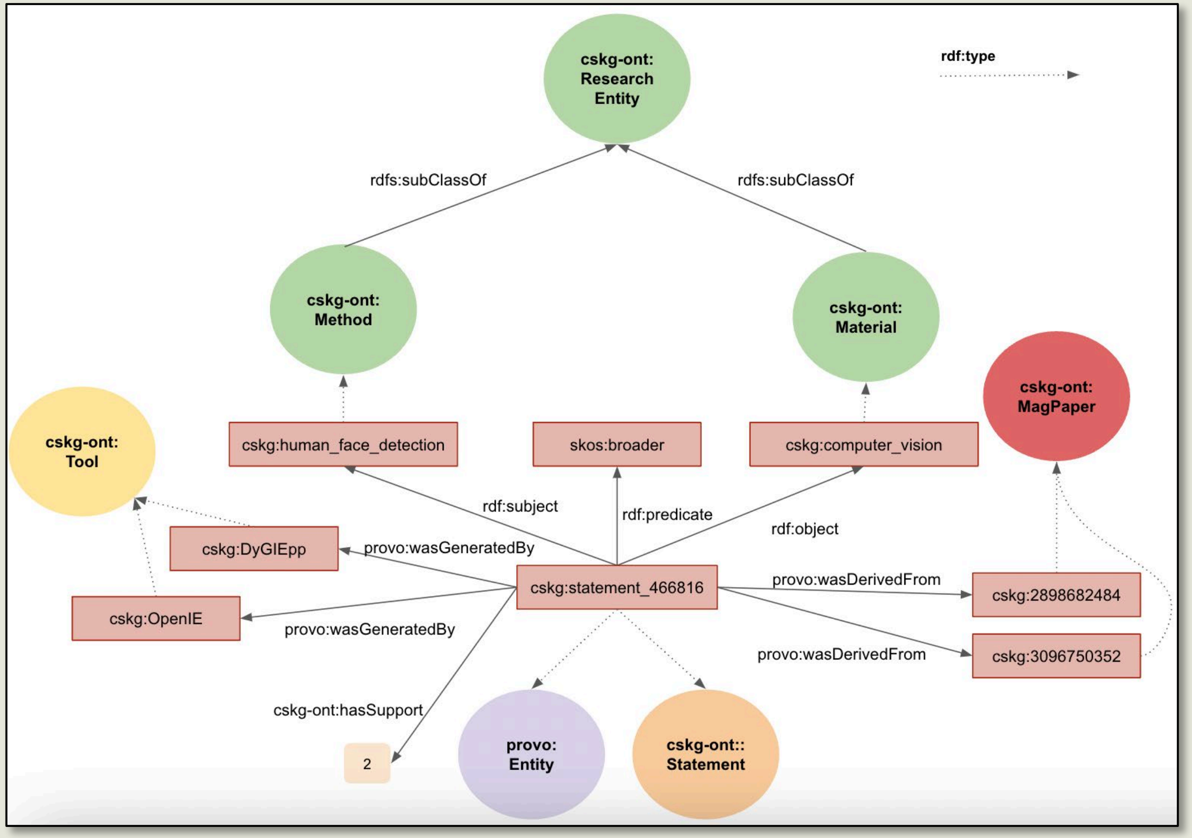
- Automatically-generated large scale examples for restricted scientific domains:
  - Artificial Intelligence Knowledge Graph (AI-KG): 1.2M statements about 820k entities from 330k papers
  - Computer Science Knowledge Graph (CS-KG): 41M statements about 10M entities/179 relations from 6.7M articles (2020-2021, currently updated every 6 months)

    https://scholkg.kmi.open.ac.uk/

Scholarly Domain KGs

- statements are claims extracted from one or more research articles in the form <subject, predicate, object>
- 5 entity types: cskg-ont:Task, cskg-ont:Method, cskg-ont:Material, cskg-ont:Metric, cskg-ont:Other
- PROV Ontology is used to track the provenance of a claim (source, processing tool that generated it)

```
cskg-ont:usesMethod rdf:type owl:ObjectProperty ;
           rdfs:subPropertyOf cskg-ont:uses ;
           rdf:type owl:TransitiveProperty ;
           rdfs:domain [ rdf:type owl:Class ;
                         owl:unionOf ( cskg-ont:Method
                                       cskg-ont:Metric
                                       cskg-ont:OtherEntity
                                       cskg-ont:Task ) ] ;
           rdfs:range cskg-ont:Method.
```



Scholarly Domain KGs

https://scholkg.kmi.open.ac.uk/sparql/

```
prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

prefix cskg: http://scholkg.kmi.open.ac.uk/cskg/resource/

prefix cskg-ont: <http://scholkg.kmi.open.ac.uk/cskg/ontology#>

SELECT (cskg:sentiment_analysis as ?sub) ?prop ?obj ?sup

FROM  http://scholkg.kmi.open.ac.uk/cskg

WHERE { ?t rdf:subject cskg:sentiment_analysis ;

 rdf:predicate ?prop ; rdf:object ?obj ;

 cskg-ont:hasSupport ?sup }

ORDER BY desc (?sup)
```

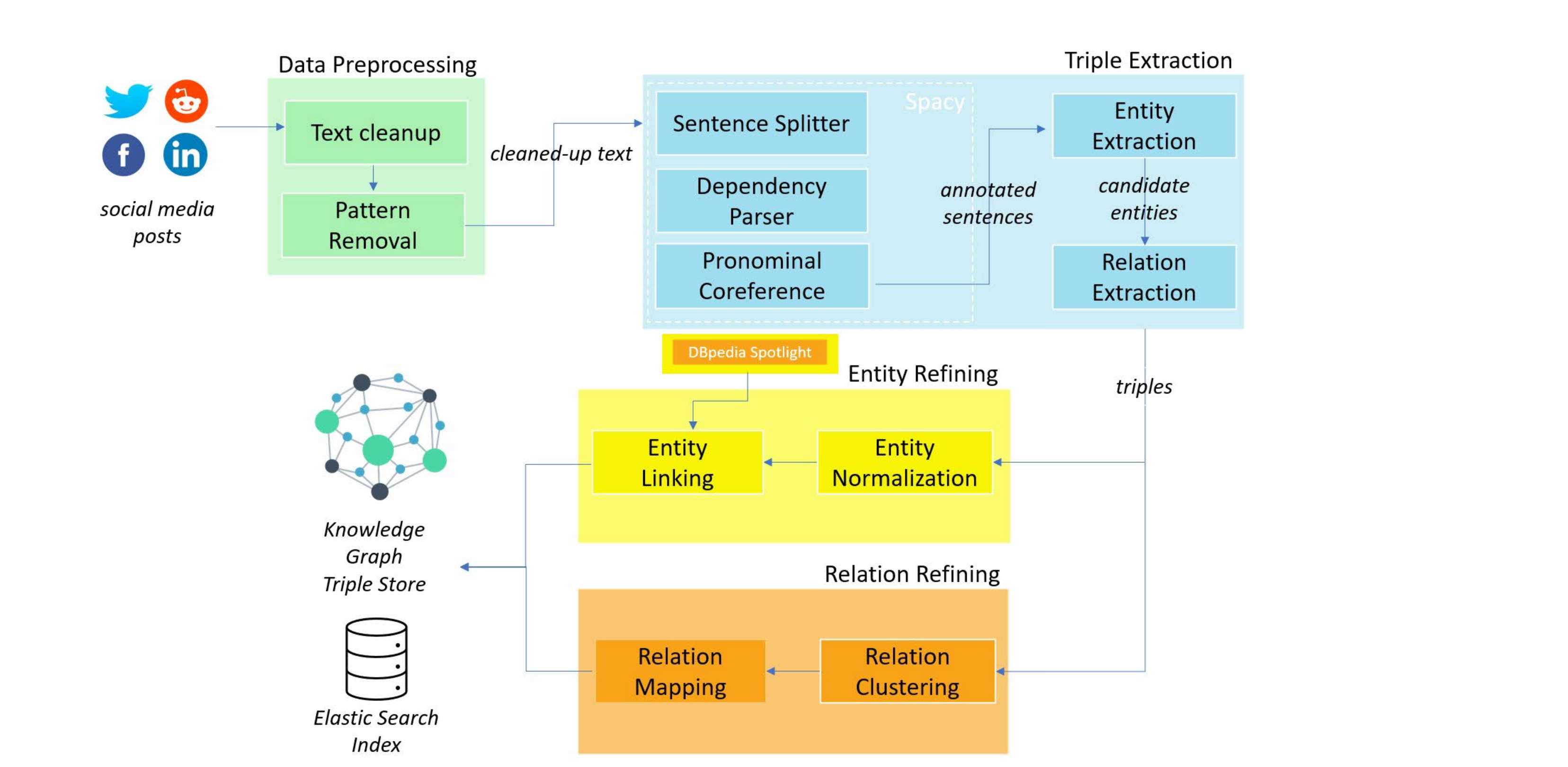
Table Response 8361 results in 2.203 seconds

	sub	prop	obj	sup
1	cskg:sentiment_analysis	<http://www.w3.org/2004/02/skos/core#broader>	cskg:natural_language_processing	"149"^^<http://www.w3.org/2001/XMLSchema#integer>
2	cskg:sentiment_analysis	cskg-ont:usesMethod	cskg:deep_learning	"83"^^<http://www.w3.org/2001/XMLSchema#integer>
3	cskg:sentiment_analysis	cskg-ont:usesMethod	cskg:machine_learning	"76"^^<http://www.w3.org/2001/XMLSchema#integer>
4	cskg:sentiment_analysis	cskg-ont:usesMaterial	cskg:twitter	"74"^^<http://www.w3.org/2001/XMLSchema#integer>
5	cskg:sentiment_analysis	cskg-ont:usesMaterial	cskg:social_social_medium	"71"^^<http://www.w3.org/2001/XMLSchema#integer>
6	cskg:sentiment_analysis	cskg-ont:usesMethod	cskg:naive_bayes	"61"^^<http://www.w3.org/2001/XMLSchema#integer>
7	cskg:sentiment_analysis	cskg-ont:usesMethod	cskg:neural_network	"59"^^<http://www.w3.org/2001/XMLSchema#integer>
8	cskg:sentiment_analysis	<http://www.w3.org/2004/02/skos/core#broader>	cskg:nlp_task	"54"^^<http://www.w3.org/2001/XMLSchema#integer>
9	cskg:sentiment_analysis	<http://www.w3.org/2004/02/skos/core#broader>	cskg:natural_language_processing_related_t...	"53"^^<http://www.w3.org/2001/XMLSchema#integer>
10	cskg:sentiment_analysis	cskg-ont:usesTask	cskg:data_augmentation	"47"^^<http://www.w3.org/2001/XMLSchema#integer>
11	cskg:sentiment_analysis	cskg-ont:usesTask	cskg:natural_language_processing	"47"^^<http://www.w3.org/2001/XMLSchema#integer>
12	cskg:sentiment_analysis	cskg-ont:usesMaterial	cskg:twitter_data_set	"47"^^<http://www.w3.org/2001/XMLSchema#integer>

Problem

- can these methods be adapted to process more fast-reactive, language varied sources such as news, micro-blogging posts?
- is there sufficient overlapping of domain entities for tracking facts/relations concerning those entities?
- testing these hypotheses: experimenting with extracting Knowledge Graphs from social media posts on a target Tech domain: Digital Transformation

Knowledge Graph generation pipeline



"Triplétoile: Extraction of Knowledge from Microblogging Text" Vanni Zavarella, Sergio Consoli, Diego Reforgiato Recupero, Gianni Fenu, Simone Angioni, Davide Buscaldi, Danilo Dessi, Francesco Osborne under review for [Information Processing & Management]

Data Collection

- **EU Projects:**

  Cordis API: 135k Horizon2020 EU project deliverables: Description+Full Text

- **Scientific Papers:**
  - OpenAlex API: 243M works, open replacement for industry-standard scientific knowledge bases (Elsevier's Scopus, Clarivate's Web of Science)
  - Semantic Scholar API: over 200M academic papers sourced from publisher partnerships, data providers, and web crawls

- **Patents:**
  - EPO's Open Patent Services (OPS) API: Up to 4 GB of data per week

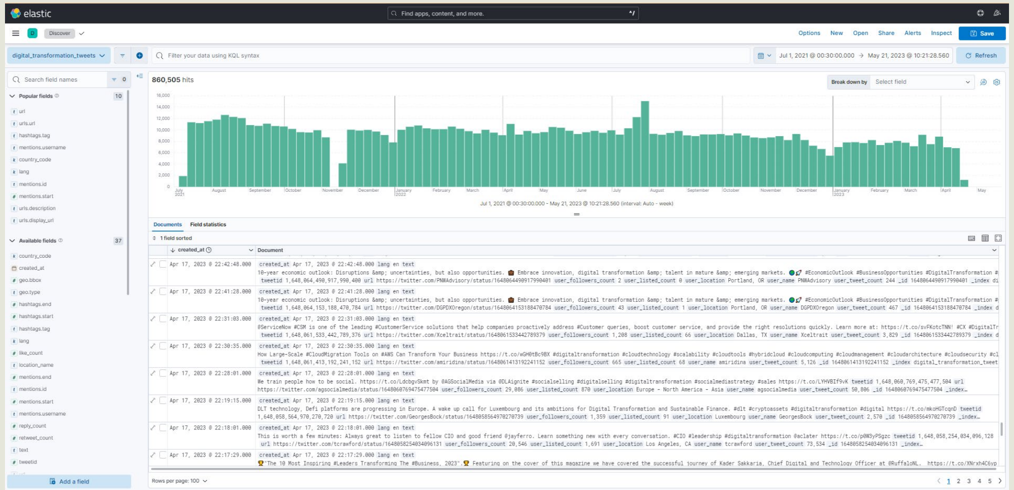
Micro-blogging text:
- Twitter/X API: Academic Access License, currently suspended: 1M tweets #DigitalTransformation dataset
- Reddit: native API

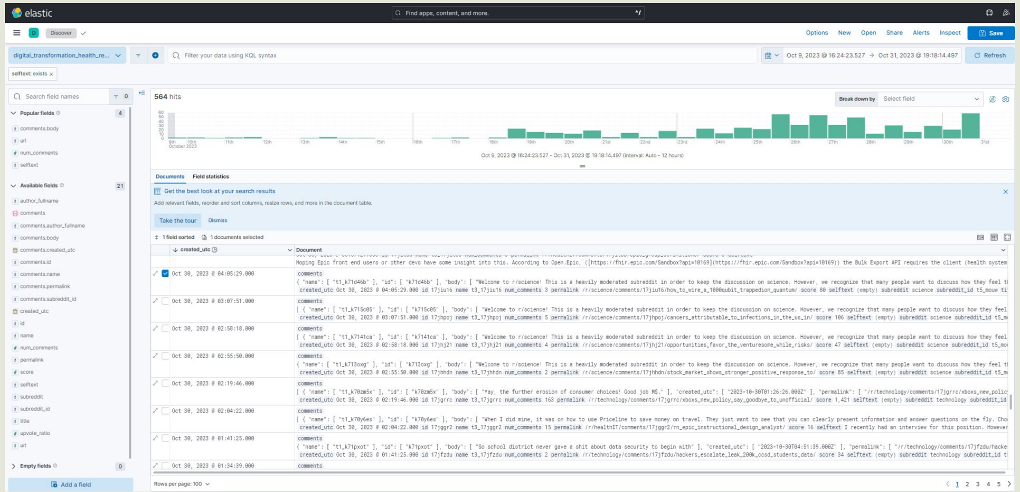
- **News:**

  using a Dow Jones Data, News and Analytics (DNA) dataset from the Joint Research Centre

Data Collection

- Collected via Twitter search API v2 a sample of ~1M English language tweets from 2002 with #DigitalTransformation (no re-tweets)
- Using a Elastic Search datalake, storing tweet text + metadata and hyperlinks
- currently collecting a Reddit thread/comment collection using Reddit native API
- Linked back from triples by tweet/thread/comment id





Data Preprocessing

- Standard NLP models struggle to process micro-blogging text

Two-fold approach:

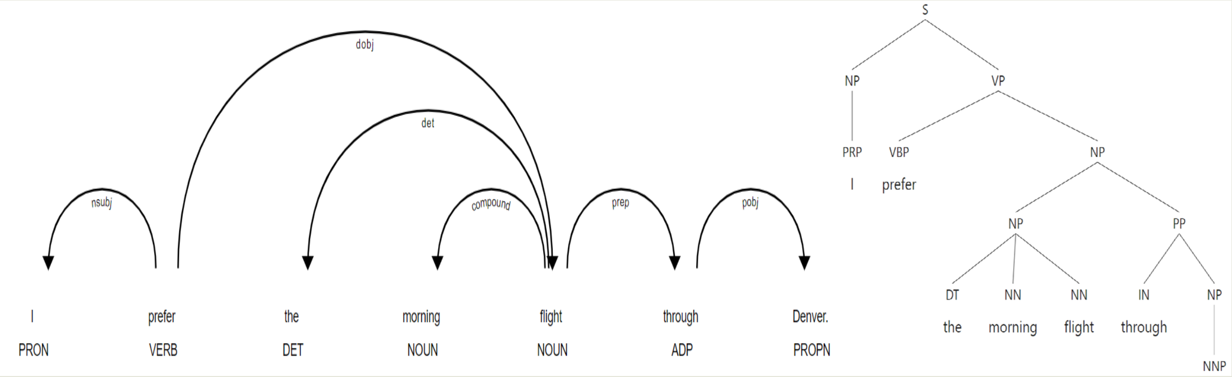
- Keep tokens and token sequences encoding platform-specific metadata carrying syntactic functions (#digitaltransformation, @NASA) and remove by default the ones which typically do not (URLs, emoticons, reserved tokens)
- Platform-specific heuristics rules to remove syntax-disruptive token patterns
    - remove sequences of n entity mentions and retweet markers at the beginning of a sentence, with $n > 1$ or when the sequence is not followed by a verb
    - or any sequence of size $n > 1$ hashtags/mentions/URL, we drop the sub-sequence with indexes $[1 : n]$ or drop the entire sequence if preceded by a sentence closing marker

Dependency parse trees

- No constituent structures, only lemmas and a set of directed, binary typed relations from **head** to **dependent**
- A directed edge-labelled acyclic graph g=(w,d) where:
  - $w \subseteq V$ (the vocabulary of the language) plus *Root*
  - One single Root with no incoming edge for each sentence
  - Any other node w has exactly 1 incoming edge
  - there is a unique path from Root to any w
- shared taxonomy of dep relations (the Universal Dependency project) valid across languages and large tree banks available for training models
- We use the Spacy's transformer *pipeline en_core_web_trf-3.6.1 (over 95% dep parsing accuracy) trained on OntoNotes*

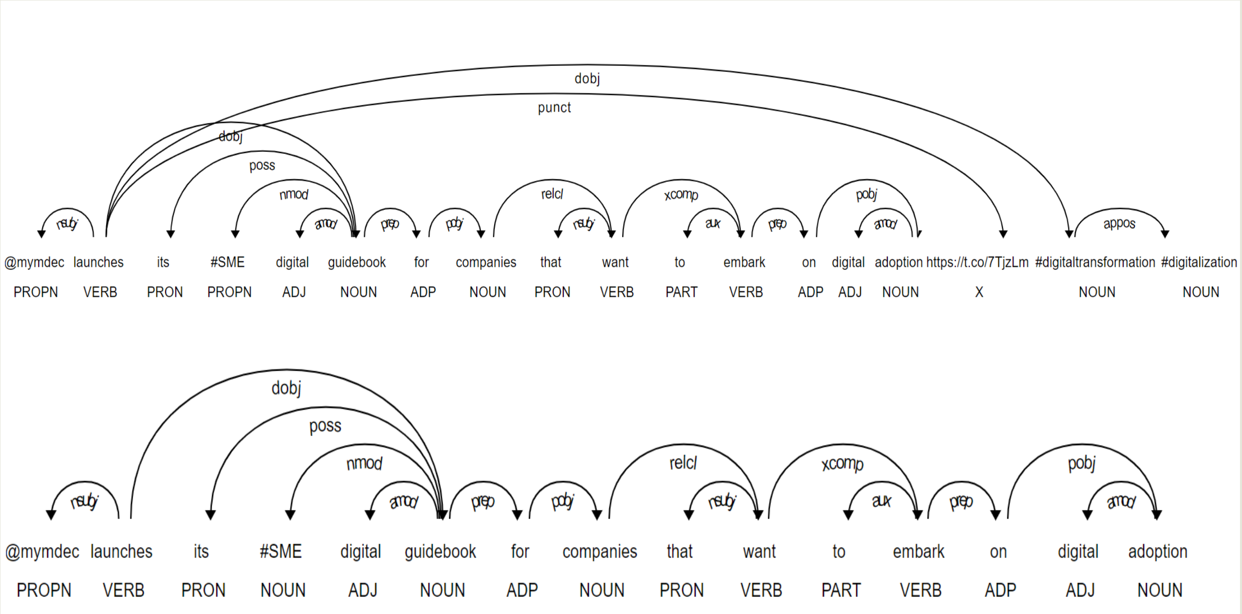
https://github.com/explosion/spacymodels/releases/tag/en_core_web_trf-3.6.1

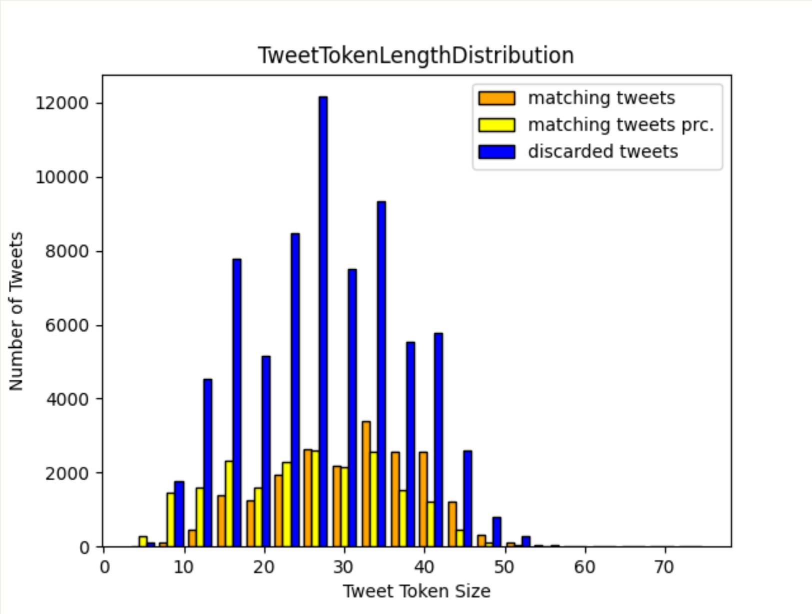
Clausal Argument Rel	Description
nsubj	Nominal subject
dobj	direct object
ccomp	Clausal complement
Nominal Modifier Rel	**Description**
nmod	Nominal modifier
amod	Adjectival modifier
Other	**Description**
conj	conjunct



Data Preprocessing

- Example fixing of parsing errors
- Text preprocessing heuristics seem to remove noise instead of information content



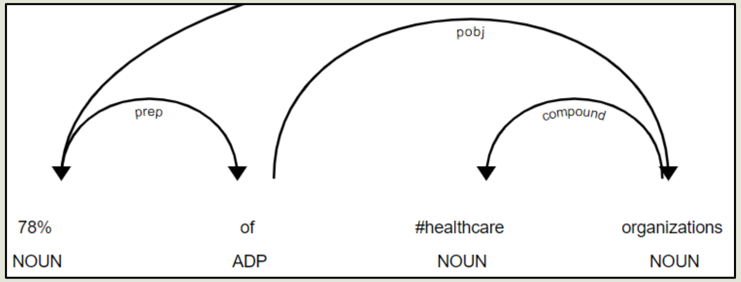


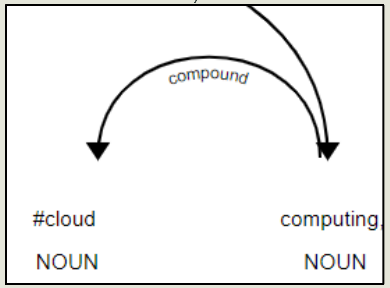
Entity Extraction

- Non-recursive patterns over Spacy dependency parse trees
- Extract and store quantitative modifiers and syntactic head
- Integrate a restricted anaphora resolution module
- **Output** is a set $E = \{e_1, \ldots e_n\}$ of unmerged candidate entity phrases

Example:

78% of #healthcare organizations are currently deploying #cloud computing, with 20% planning to deploy it in the future.





Entity Refining

- **Goal**: clean up and normalize the candidate entities into a form that allows the merging across entity name variants
- Normalization: e.g. "#SmartCities" → "smart cities", etc.
- Feed the normalized text as input to the Spacy's DBpedia Spotlight model and link to DBpedia KG (*owl:sameAs*) the original entities if they (a.) contain **and** (b.) share syntactical head with Spacy entity spans
- Otherwise we draw a weaker 'relatedness' link (*skos:related*)

Candidate Entity	Canonical Form	Linked DBpedia Entity	Related DBpedia Entity
78% of #healtcare organisations	**Form**: *healthcare_organisation* **Head**: *organisation* **Quant**: 78%	-	http://DBpedia.org/resource/Health_care
#digitaltransformation leaders	**Form**: *digital_tranformation_leader* **Head**: *leader*	-	http://DBpedia.org/resource/Digital_Transformation
Gartner	**Form**: *gartner* **Head**: *gartner*	http://DBpedia.org/resource/Gartner	http://DBpedia.org/resource/Gartner
@Gartner_inc	**Form**: *gartner_inc* **Head**: *gartner_inc*	http://DBpedia.org/resource/Gartner	http://DBpedia.org/resource/Gartner
Gartner survey	**Form**: *gartner_survey* **Head**: *survey*	-	http://DBpedia.org/resource/Gartner

Relation Extraction

- **Goal**: generate a set of candidate verbal relations $V = v_0, \ldots, v_k$ and a set of triples $S = s_0, \ldots, s_k$ of the form $< e_m, v, e_n >$ where $v \in V$ and $e_m, e_n \in E$
- Method:

```
For each dep tree G_s of sentence s
     for each pair of candidate entities e_m, e_n in s
               collect all shortest paths p in G_s connecting e_m, e_n such that:
                        p contains a verb node v
                        p is in a verified pattern list VP
```

- VP was filtered using majority voting among 3 experts from the 20 most frequent of a set of 3695 path shortest patterns connecting automatically annotated entities in a separated corpus

Target Dependency Paths
[nsubj,dobj]
[acl,relcl,dobj]
[acl,dobj]
[nsubjpass,agent,pobj]
[nsubj,dobj,conj]
[nsubj,conj]

Sample Discarded Paths
[obj,pobj]
[obl,pobj]
[nsubj,pobj,nmod]

Relation Refining

- **Goal**: generalize from the set $S = s_0, \ldots, s_k$ of surface form triples of type $< e_m, v, e_n >$ to the lower sized set $T = T_0, \ldots, T_k$ of triples of the form $< \varepsilon_m, r, \varepsilon_n >$ where each $\varepsilon_i \in \mathrm{E}$ is an entity and r is a label in a common relation vocabulary R
- Derive relation embeddings vectors
- Apply dimensionality Reduction and Clustering
- Mapping relation verbs to cluster representatives

Subject Entity	Relation	Object Entity	Support
pandemic	accelerate	digital_transformation	15
artificial_intelligence	impact	insurance_sector	7
microsoft	buy	riskiq	6
data-driven_insight	drive	decision-making	5
agile_business	demand	effective_marketing_capability	4
hootsuite	buy	ai_chatbot_firm	4
automl	generate	data-driven_insight	2
image_classification	use	transfer_learning	2
new_belgium_brewing	implement	digital workflow_place_solution	2
e-rupi	back	existing indian rupee	1
82%_of_cio	implement	new_technology	1
image_recognition_framework	use	artificial_intelligence	1
microinsurance	close	africa_insurance_gap	1
hsbc_qatar	introduce	mobile_payment	1
ford_motor_company	explore	blockchain_technology	1

Relation Embeddings

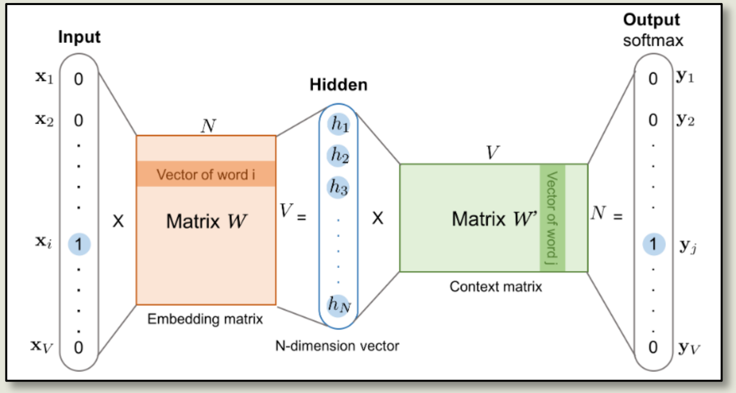
- Starting with a set of 29,335 raw triples, we derived 2,539 unique 300-dimensional word embeddings from GloVe and standardized them
- non-contextual embeddings from Spacy's *en_core_web_lg-3.6.0* LM
- GloVe architecture: Shallow NN, simplification of predictive language models like Word2Vec skip-gram, gradient descent minimizes the cost function:

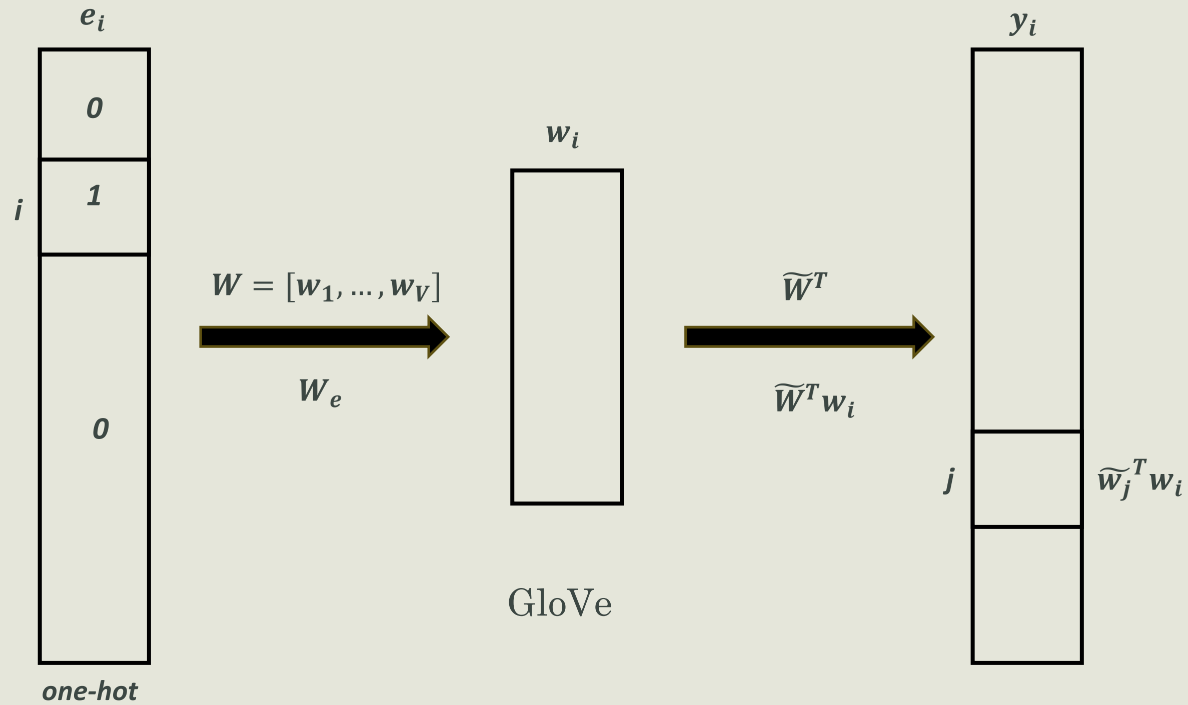
$$J = \sum_{i,j=0}^{V} f\left(X_{i,j}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{i,j}\right)^2$$

V is the vocabulary size

$X_{i,j}$ = count of i-th and j-th words co-occurring in a window

Word2Vec
Skyp-gram





Relation Clustering

- HDBSCAN is a hierarchical version of the popular density-based DBSCAN algorithm
- sound assumptions for our use case:
  - does not require to preset the number of clusters
  - it considers outliers and leaves un-clustered the data points lying in low-density regions

Problem: high dimensional data require more observed samples to produce the suitable level of density for HDBSCAN to work properly

Solution: applying UMAP to perform non-linear dimension reduction the dataset dimension gets small enough for HDBSCAN to cluster most of the instances

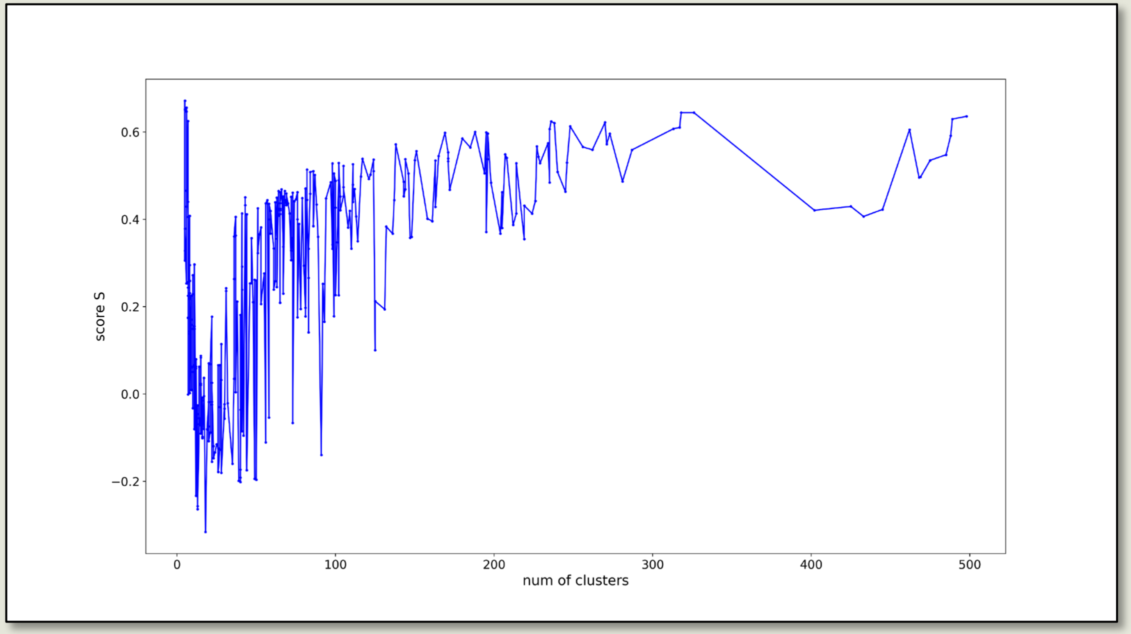
Relation Clustering

- optimize the UMAP-HDBSCAN combination by grid search over the hyperparameters
- We define a target score: $S = silhouette_X \cdot clustered_X$
  - $silhouette_x$ of an instance $x \in X$ is equal to: $\frac{b-a}{\max(a,b)}$ with a being the mean distance to the other instances in the same cluster, and b being the mean distance to the instances of the next closest cluster
  - $clustered_x$ is the is the fraction of instances of X that were actually clustered by HDBSCAN

HDBSCAN		
`min_cluster_size`	smallest data point groupings that are considered as clusters	[3,**5**,10,15]
`min_samples`	number of samples in a neighbourhood for a point to be considered a core point	[None, 1, **3**]
`cluster_selection_epsilon`	distance threshold under which clusters will be merged	[**0.0**, 0.2, 0.5]
UMAP		
`min_dist`	controls how tightly UMAP is allowed to pack points together	[**0.0**, 0.1]
`n_neighbors`	how many data points UMAP is looking at when attempting to learn the manifold structure of the data	[**5**, 10, 50]
`n_components`	dimensionality of the reduced space to embed data into	[2, **3**, 5, 10,20]

Relation Clustering

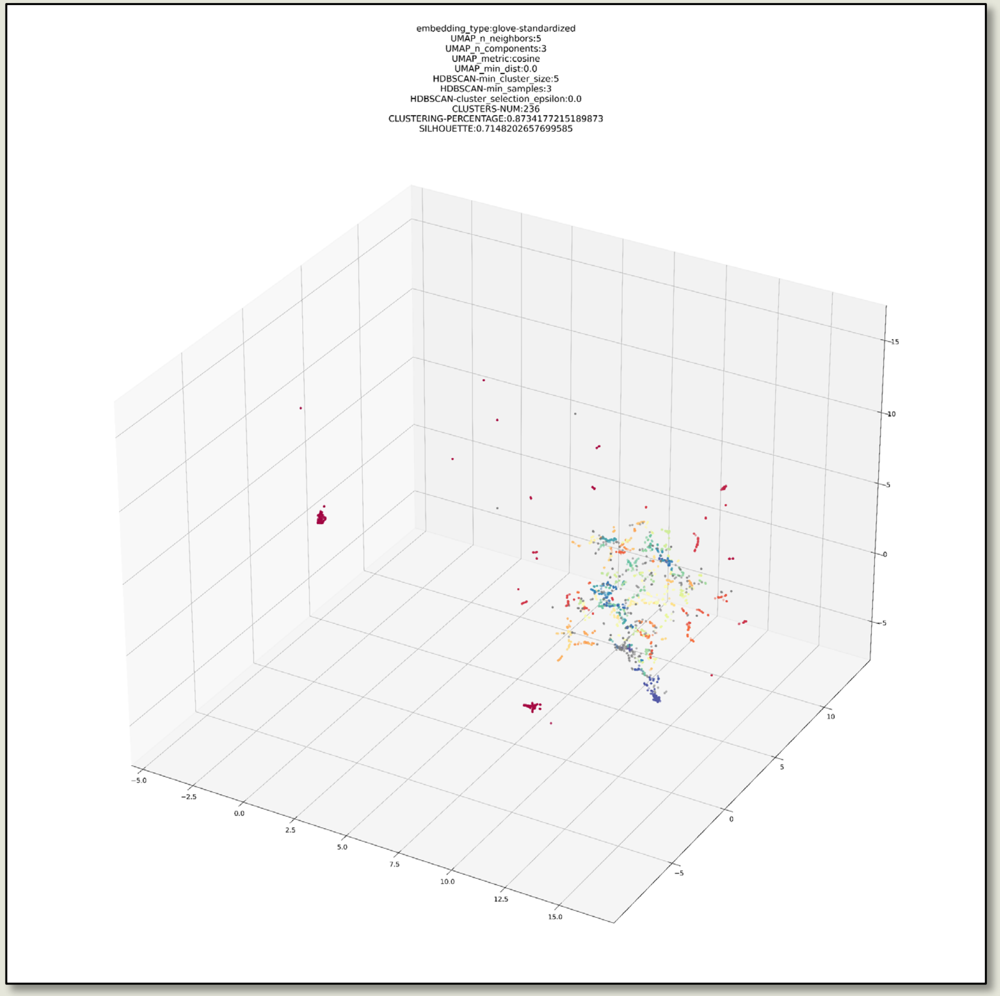
- Select a subset of best-scoring UMAP-HDBSCAN configurations and plotted their S score over the number of output clusters they generate
- pick a sub-optimal configuration that balances between generalization (fewer clusters) and accuracy (cluster number closer to the dataset size)
- overall score of around 0.62, silhouette score on clustered points 0.71 and data clustering percentage 0.87, returning 236 clusters, with an average cluster size of 12 elements



Relation Clustering

- UMAP-computed 3-dimensional space representation of the relation embedding vectors for the chosen clustering configuration

- relatively local structure is accurately captured, with few data points left un-clustered (marked in grey)



Relation Mapping

- Finally, for each relation verb v in the dataset, we replace it with the predicate label r consisting of the lemma of the most frequent relation in the cluster of v.
- If not clustered, we map it to itself

Relation Verb	Relation Predicate	Example
fuel	FUEL	*'How the UR+ Ecosystem is fueling Cobot Market Growth'*
driven by	FUEL	*'Digital transformation in Ho Chi Minh is being driven by remote working'*
accelerated by	FUEL	*'"huge social trends being accelerated by the pandemic'*
identify	IDENTIFY	*'Machine learning can identify signs of Alzheimers in patients'*
quantify	IDENTIFY	*'Research quantifies G's potential in roaming and manufacturing'*
predict	IDENTIFY	*'AI-supported test can predict eye disease that leads to blindness'*

Evaluation

- Human expert assesment: 500 statements, equally distributed among high-support (>= 5) and low-support triples
- Annotators were instructed to assign True if
  - the subj and obj entities are linked by a relation in the tweet text
  - the assigned relation label entails the relation verb in the tweet text
  - the spans of the subject/object of extracted triples include the syntactic head of the relation's subject/object
- 3 evaluators with majority vote (Fleiss K_F agreement = 0.558, substantial agreement/ pairwise Cohen K agreement = 0.61)
- overall **Precision** of **0.96**, individual rates ranging from 0.90 to 0.96
- Primary error sources: failure in the syntactic parsing of the sentence, inaccuracy of relation clustering/mapping error in pronominal anaphora resolution

Evaluation

- Comparative Evaluation: on 500 random tweets we run our pipeline and merged extracted candidate entities with the one generated by the DyGIE++ Extractor
- run 4 alternative methods to identify relationships between these entities from the same set of tweets
- measured number of extracted triples (approximation to recall when combined with Precision estimate)
- Human expert majority vote Precision assessment of 150 triple sample (Fleiss K_F agreement = 0.86)
- significant advantage over the Dependency-based Extractor method, which deploys very similar syntactic information from the sentence (may be due to the application of the processing step upstream)

Extraction Method	Generated Triples	Precision
OpenIE Extractor	588	0.52
PoST Extractor	**1015**	0.17
Dependency-based Extractor	339	0.77
Entity and Relationship Refiner	348	0.31
Triplétoile	663	**0.82**

D. Wadden, U. Wennberg, Y. Luan, H. Hajishirzi, Entity, relation, and event extraction with contextualized span representations

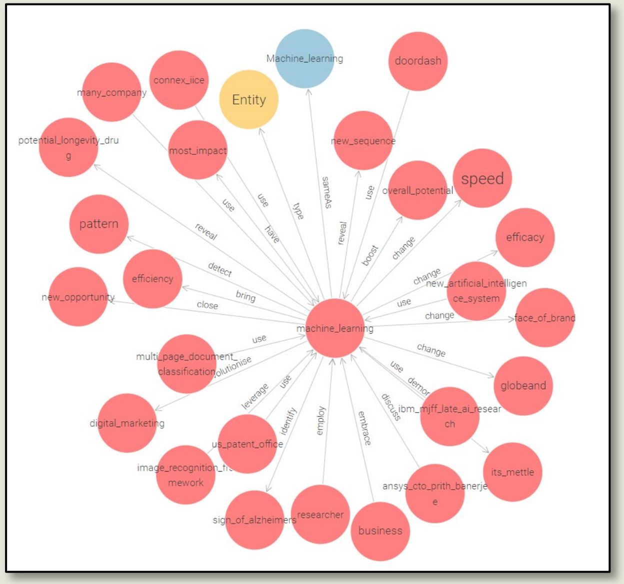
Triple Store: DTSMM

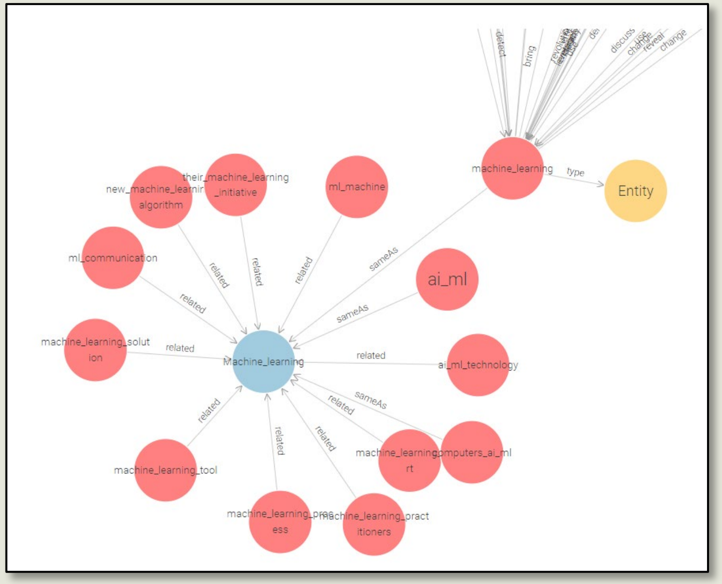
- First prototype 22270 triple store extracted from the test 100k tweet sample
- Reification of claims into dtsmm-ont:Statement class instances, encoding support, provenance and negation attributes

```
dtsmm-ont:statement_10100 a dtsmm-ont:Statement,
rdf:Statement ;
dtsmm-ont:negation false ;
dtsmm-ont:comesfromTweet dtsmm:tweet_1424266328882429952 ;
...
dtsmm-ont:hasSupport 6 ;
rdf:subject dtsmm:multi_page_document_classification ;
rdf:predicate dtsmm-ont:use ;
rdf:object dtsmm:machine_learning .
```

- DTSMM provides 2,857 owl:sameAs links and 3,309 skos:related links to DBpedia entries

Triple Store: DTSMM





- Triple store is currently being updated with Reddit and News data
- A more mature version of the Knowledge Graph will be made publicly accessible as Terse RDF dump under the ‘Dataset Socioeconomic Tracker using Unconventional Data’ within the EU Data Portal:

  https://data.jrc.ec.europa.eu/dataset/f7be47f7-49a2-44e8-9dc8-043735af4139

- The direct link to the Digital Transformation knowledge graph, available in Terse RDF Triple Language (Turtle):

  https://jeodpp.jrc.ec.europa.eu/ftp/jrc-opendata/CC-COIN/se-tracker/DTSMM_KG.ttl

Current Limitations

- **Data collection**: native fine-grained topic classification not available for news/SM post, so automatic sampling methods are needed to increase recall
- **Entity/Relation Extraction**: does not rely on the ontology specification of a target domain in order to customize the extraction process
- **Relation Mapping**: a domain-specific classification schema for relations would allow setting up a supervised learning of the relation mapping
- Current low scale prevents using inductive graph learning methods

Ongoing Developments

- **Data collection**: using transformer-based topic classification method (SBert) for collecting more accurate sample of input data
- Integrating with KGs from more ‘standard’ sources
- adapting existing supervised learning framework (e.g. DyGIE++):
  - categorize unlinked entities
  - categorize relations

THANK YOU!

zavavan@yahoo.it