

Detecting fake news using machine learning and reasoning in Description Logic

Adrian Groza and Aron Katona

Department of Computer Science
Technical University of Cluj-Napoca, Romania
Adrian.Groza@cs.utcluj.ro



Myth #1: The day in which you were born guides you to choose a profession

best suited for your abilities. It also *increases your chances of success* in the career.

Example (Those born on 1st, 10th, 19th, 28th)

The ruling planet for *number 1* is the Sun. They are born *leaders*. They are risk takers with a 'never surrender' attitude, which makes them excel in *business*. Elon Musk and Bill Gates belong to these dates. **Business** is the best career for number 1 individuals.

Example (Those born on 2nd, 11th, 20th, 29th)

They are *creative* people. The best suitable career options for them is **diplomacy, arts, acting, fashion designing**. Jennifer Aniston and Leonardo Di Caprio were born on these dates.

- 1 Pick classes from YAGO/DBPEDIA, representing successful *careers* according to the myth: *leader, political leader, executive director, managing director*.
- 2 Calculate the **proportion** p_1 of people born in days 1, 10, 19 and 28 that are part of the chosen classes. Find the sample size n_1 .
- 3 Calculate the **proportion** p_2 of people born in the other days that are part of the chosen classes. Find the sample size n_2 .
- 4 Compare the proportions. Check how is the difference compared to $\delta = 15\%$. Calculate p-value to find out if the results are relevant

How many leaders were born on day 10?

```
1 SELECT COUNT(?person) as ?nr_persons
2 WHERE { ?person yago:wasBornOnDate ?date ;
3         rdf:type ?person_type ;
4         rdf:type ?class .
5         ?class rdfs:label "leader"@eng
6         ?person_type rdfs:label "person"@eng .
7 FILTER (datatype(?date) = xsd:date
8 && fn:month-from-date(xsd:date(?date)) = 7
9 && fn:day-from-date(xsd:date(?date)) = 10) }
```

Career	n_1	n_2	p_1	p_2	
leader			0.00848	0.00832	1.89×10^{-46}
political leader	1,660,382	10,441,109	0.00292	0.00278	6.27×10^{-10}
executive director			0.00122	0.00116	5.25×10^{-5}
managing director			0.00148	0.00145	4.95×10^{-10}
artist			0.004274	0.00437	2.84×10^{-41}
actor	1,546,551	10,554,940	0.003862	0.003834	5.6×10^{-25}
diplomat			0.000361	0.000348	0.00778

Myth #2: The 'life path number' tells us if two persons are compatible

Repeatedly summing all the digits in one's birth date (e.g. 01.02.1998 $\rightarrow$ 2)

Example (LPN=1 compatible 3, 5, and 6)

Number 1's can sometimes be a little bossy and opinionated. 3 and 5 can put up with this sort of character as they have traits that can help them deal with this easier. [...]

Example (LPT=2 compatible 6, 8, and 9)

LPT=2, you are a natural peacemaker and will often see all sides of a story. You find it easy to deal with difficult situations too. This makes you very compatible with the number 8, who can sometimes become quite indecisive at times.

- 1 Get the dates of birth for each married couple in DBpedia
- 2 Calculate the proportion p_1 of people with $lpn = 1$ which are married to someone with $lpn \in \{3, 5, 6\}$
- 3 Calculate the proportion p_2 of people with $lpn \notin \{1, 3, 5, 6\}$ which are married to someone with $lpn \in \{3, 5, 6\}$
- 4 Compare the proportions. Check if the result is larger than a threshold (15%)
- 5 Calculate p-value

Retrieving the birth dates of each married couple

```
1 SELECT ?date1, ?date2
2 WHERE { ?person1 rdf:type ?person_type;
3         dbo:birthDate ?date1;
4         dbo:spouse ?person2.
5         ?person2 rdf:type ?person_type;
6         dbo:birthDate ?date2.
7         ?person_type rdfs:label "person"@en.}
```

l_{pn}	'compatible' l_{pns}	n_1	n_2	p_1	p_2	
1	3, 5, 6	3,316	25,837	0.3543	0.3638	2.06×10^{-13}
2	6, 8, 9	3,228	25,925	0.3782	0.3767	6.01×10^{-10}
3	1, 5, 7	3,011	26,142	0.3251	0.2996	0.0155
4	1, 8, 7	3,400	25,753	0.3321	0.3507	6.41×10^{-17}
5	1, 3, 7	3,283	25,870	0.3290	0.3340	1.26×10^{-10}
6	1, 2, 8, 9	3,199	25,954	0.4495	0.4432	5.27×10^{-11}
7	3, 5	3,223	25,930	0.2144	0.2132	2.97×10^{-5}
8	2, 4, 6	3,332	25,821	0.3190	0.3293	1.81×10^{-12}
9	2, 6	3,161	25,992	0.2186	0.2127	0.000418

Semantic Web Vision

"The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation."



Tim Barners Lee



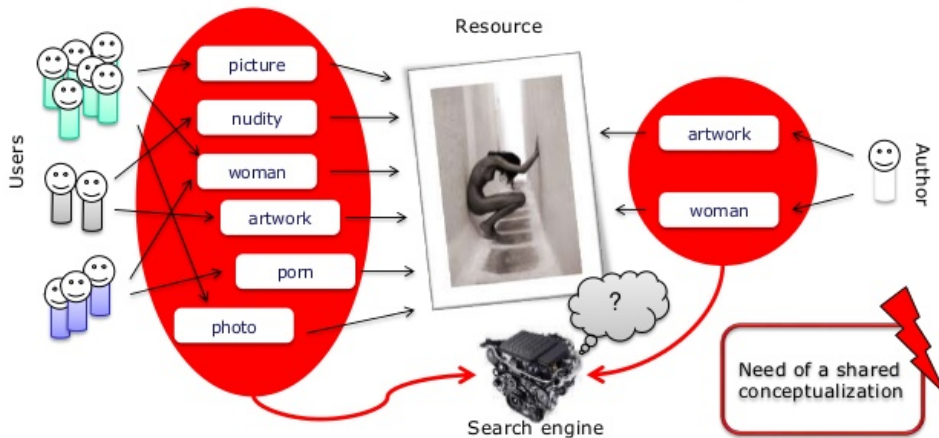
Social Web



Semantic Web

collaborative tagging
Web 2.0 approach

authoritative metadata
Semantic Web approach



Conceptualization

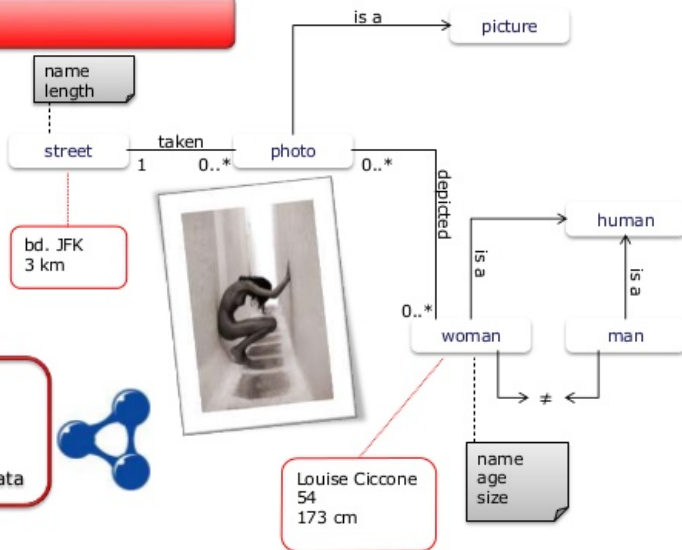
concepts

relations between
concepts

attributes

instances

Using the same
ontology allows two
different systems to
communicate and to
reason over (meta)data



Myth

5G mobile networks spread Covid-19

Exposing yourself to temperatures higher than 25C degrees prevents COVID

Drinking excessive amounts of water can flush out the virus

Regularly rinsing your nose with saline help prevent infection with Covid-19

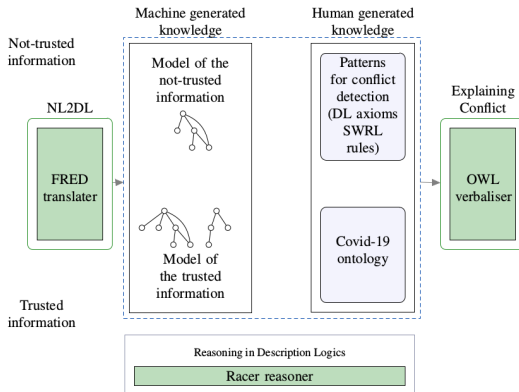
Eating raw ginger counters the coronavirus

Antibiotics are effective in preventing and treating the new coronavirus

High dose of Vitamin C heals Covid-19

The new coronavirus can be transmitted through mosquito

Covid-19 can affect elderly only



Example (Terminological box)

"Coronavirus disease (COVID) is an infectious caused by a newly discovered coronavirus"

$COVID \equiv CoronavirusDisease$ (1)

$InfectiousDisease \sqsubseteq Disease$ (2)

$CoronavirusDisease \sqsubseteq InfectiosDisease \sqcap \forall causedBy.NewCoronavirus$ (3)

Example (Assertional Box)

$SARS-CoV-2 : Virus$ (4)

$hasSource(SARS-CoV-2, bat)$ (5)

Example (Incoherent ontology)

$Covid-19 \sqsubseteq InfectionDisease$ (6)

$Covid-19 \sqsubseteq \neg InfectionDisease$ (7)

Example (Reasoning to detect incoherence)

$Covid-19 \sqsubseteq InfectionDisease$ (8)

$InfectiousDisease \sqsubseteq Disease \sqcap \exists causedBy.(Bacteria \sqcup Virus \sqcup Fungi \sqcup Parasites)$ (9)

$Covid-19 \sqsubseteq \neg Disease$ (10)

Example (OIL antipattern)

$OIL_1 : Antibiotics \sqsubseteq \forall kills.Virus$ (11)

$OIL_2 : Antibiotics \sqsubseteq \forall kills.Bacteria$ (12)

$OIL_3 : Virus \sqsubseteq \neg Bacteria$ (13)

Myth: 5G mobile networks spread Covid-19

$5G : \text{MobileNetwork}$ (14)

$\text{covid19} : \text{Virus}$ (15)

$\text{spread}(5G, \text{covid19})$ (16)

Fact: Viruses can not travel on radio waves/mobile networks

$\text{Virus} \sqsubseteq \neg(\exists \text{travel}.(\text{RadioWaves} \sqcup \text{MobileNetworks}))$ (17)

Detecting inconsistencies

$\text{Virus} \sqsubseteq \neg(\exists \text{travel}.\text{MobileNetworks})$ (18)

$\text{Virus} \sqsubseteq \forall \text{travel}.\neg \text{MobileNetworks}$ (19)

$\text{Virus} \sqsubseteq \forall \text{spread}.\neg \text{MobileNetworks}$ (20)

Here we need the subsumption relation between roles ($\text{travel} \sqsubseteq \text{spread}$). The reasoner finds that the individual 5G (which is a mobile network by axiom (14)) that spreads COVID19 (which is a virus by axiom (15)) is in conflict with the axiom (20).

Myth: Covid-19 can affect elderly only

$$\text{Covid-19} \sqsubseteq \forall \text{affects.Elderly} \quad (21)$$

Fact: Covid-19 can affect anyone

$$\text{Covid-19} \sqsubseteq \forall \text{affects.Person} \quad (22)$$

Detecting inconsistency: Abox + background knowledge

$$\text{affectedBy}(\text{jon}, \text{Covid-19}) \quad (23)$$

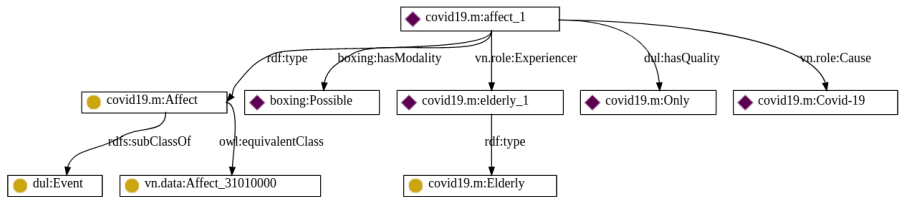
$$\text{hasAge}(\text{jon}, 40) \quad (24)$$

$$\text{Elderly} \sqsubseteq \text{Person} \sqcap (> \text{hasAge } 65) \quad (25)$$

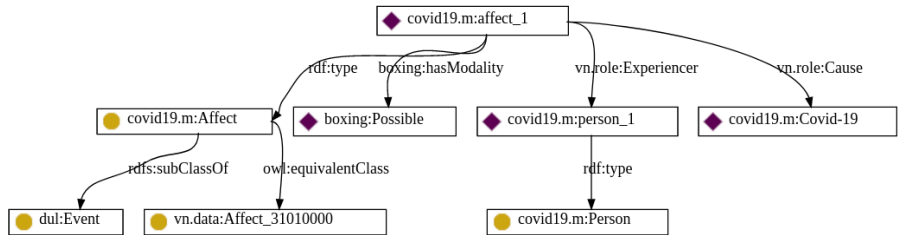
$$\text{affects}^- \equiv \text{affectedBy} \quad (26)$$

$$\text{Covid-19} \equiv \text{one} - \text{of}(\text{Covid-19}) \quad (27)$$

Step 1: “Covid-19 can affect elderly only”



Step 2: “Covid-19 can affect anyone”



Step 3: Merge ontologies

Step 4: Add background knowledge

Step 5: Detect inconsistency/incoherence

Step 6: Verbalise the inconsistency

Covid-19 $\sqsubseteq \forall affects. Person$ can be learned from data

Learning description logic concepts

Example (Father example: Background knowledge)

$Male \sqsubseteq \neg Female$

markus : Male, *heinz* : Male, *stefan* : Male, *martin* : Male

anna : Female, *michelle* : Female

$\langle markus, anna \rangle : hasChild$, $\langle anna, heinz \rangle : hasChild$

$\langle stefan, markus \rangle : hasChild$, $\langle martin, heinz \rangle : hasChild$

Example (Training examples)

Positive: *stefan*, *markus*, *martin*

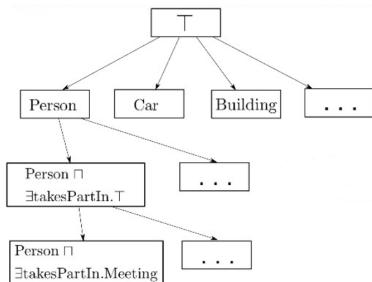
Negative: *heinz*, *anna*, *michelle*

Results (CELOE algorithm)

Solution	Father	Accuracy	F-measure
1	$Male \sqcap \exists hasChild.\top$	1	1
2	$\neg Female \sqcap \exists hasChild.\top$	1	1
3	$Male \sqcap (Female \sqcup \exists hasChild.\top)$	1	1
4	$Female \sqcup (\exists hasChild.\top) \sqcap \neg Female$	1	1
5	$Male \sqcap (\neg Male \sqcup \exists hasChild.\top)$	1	1
6	<i>Male</i>	0.83	0.86
7	$\neg Female$	0.83	0.86

Class Expression Learner for Ontology Eng.

- Generate and test: generates a set of class expressions C_i ordered based on an heuristic (Inductive Logic Programming)
- Semi-automatic approach to add axioms of the form $Target \equiv C_i$, $Target \sqsubseteq C_i$
- Combination of a refinement operator and a search algorithm (very large search space)
- Top-down search



$T \rightsquigarrow Disease \rightsquigarrow Disease \sqcap \exists hasSymptom.T \rightsquigarrow Disease \sqcap \exists hasSymptom.Fever$

Dataset (<https://github.com/nshomron/covidpred>): 14,729 (+) and 260,227 (-)

● Covid-19_Symptom SubClassOf Symptom

● female SubClassOf Person

● male SubClassOf Person

● Person DisjointWith Symptom

● female DisjointWith male

◆ cough Type Covid-19_Symptom

◆ fever Type Covid-19_Symptom

◆ head_ache Type Covid-19_Symptom

◆ shortness_of_breath Type Covid-19_Symptom

◆ sore_throat Type Covid-19_Symptom

■ hasSymptom Domain Person

■ hasSymptom Range Symptom

◆ person-97 Type contact_with_confirmed

◆ person-97 Type male

◆ person-98 Type Person

◆ person-98 Type age_60_and_above

◆ person-98 Type male

◆ shortness_of_breath Type Covid-19_Symptom

◆ sore_throat Type Covid-19_Symptom

◆ person-1 hasSymptom cough

◆ person-14 hasSymptom cough

◆ person-14 hasSymptom fever

Example (Natural language counter speech)

From input text: SARS-CoV-2 is a Bacterium.

From trusted sources (counterspeech):

1. No Virus is a Bacterium.

(<https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/myth-busters>)

2. SARS-CoV-2 is a Virus. (<https://www.cancer.gov/publications/dictionaries/cancer-terms/def/sars-cov-2>)

Conclusion

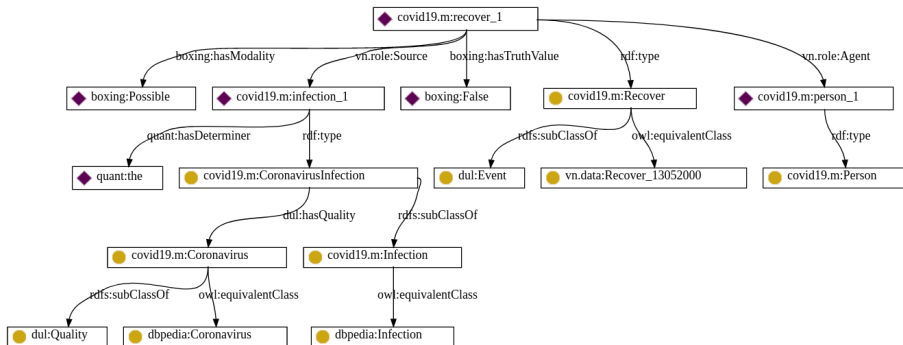
- We built an automated fact checker with explanations
- The trusted knowledge can be:
 - 1 imported from the Semantic Web
 - 2 generated from NL (based on FRED)
 - 3 learned from data (based on DL-Learner and inductive logic programming); human-in-the-loop
- Detecting fake news triggers a counterspeech generation



$\exists \text{participates}.\{thisSession\} \sqsubseteq \exists \text{thank}^-. \{I\}$

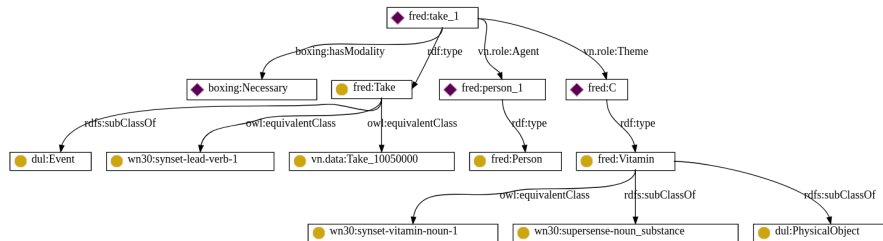
Formalising negation

You can not recover from the coronavirus infection



Modalities

You should take vitamin C



Test sentences

#	Sentence	Verdict	Comment
1	Viruses are not bacteria.	true	Type Negation Pattern - plural
2	A virus is not a bacterium.	true	Type Negation Pattern - singular
3	Bacteria are viruses.	false	Positive form of (2), incoherent, DS
4	Antibiotics kill viruses.	false	VerbNet Role Pattern: Agent, Patient; inconsistent
5	Bacitracin kills viruses.	false	(4) + WordNet
6	Amoxicillin kills retrovirus.	false	(4) + WordNet
7	Amoxicillin kills retrovirus. Bacteria are not viruses.	false	Composed: (1) + (6), with entailed axioms
8	Covid-19 is caused by SARS-CoV-2.	true	VerbNet Role Pattern: Theme1, Theme2
9	Covid-19 is caused by a coronavirus.	true	(8) + concept matching
10	Covid-19 is caused by a bacterium.	false	(9) + inconsistency
11	SARS-Cov-2 is a virus and a bacterium.	false	and
12	SARS-CoV-2 is a virus or a bacterium or an infection.	true	Type disjunction pattern
13	Vitamin C heals Covid-19.	false	VerbNet Role Pattern, inconsistent
14	Sars-CoV-2 is a virus.	true	
15	Sars-CoV-2 is not a virus.	false	Type Negation Pattern, instance matching
16	Sars-CoV-2 is not a Bacterium.	true	Type Negation Pattern, instance matching