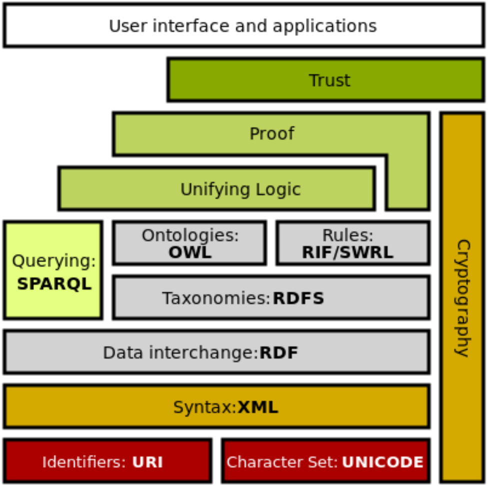
Introduction to ontologies in computational biology

Robert Hoehndorf

Overview

The Semantic Web



Web Ontology Language (OWL)

- OWL 2 is based on the Description Logic $\mathcal{SROIQ}(\mathcal{D})$
- $\mathcal{ALC}$ with
  - complex role inclusions: $r \circ s \subseteq r$
  - role hierarchy: $r \subseteq s$
  - role transitivity $r \circ r \subseteq r$
  - nominals: $\{a_1, ..., a_n\}$ as concept constructor
  - qualified number restrictions: $(\leq nr.Q)$
  - datatype properties: $\exists r.[\geq n(Integer)]$

Terminology

- Instances
- Properties
  - Object properties
  - Datatype properties
- Classes
- Meta-classes
  - OWL Full
  - Punning
- Axiom
  - Class axioms: Subclass, Equivalent class, Disjoint class
  - Property axioms
- Ontology
- OWL: Web Ontology Language

Syntax

- originally an extension of RDF and RDF Schema
- several different syntaxes

Consider the axiom $Parent \equiv Human \sqcap \exists hasChild.\top$

Functional Syntax

```
EquivalentClasses(:Parent
  ObjectSomeValuesFrom(:hasChild owl:Thing))
```

RDF/XML Syntax

```
<owl:Class rdf:about="http://example.com/demo-ontology.owl#Parent">
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://example.com/demo-ontology.owl#hasChild"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
        </owl:Restriction>
    </owl:equivalentClass>
</owl:Class>
```

RDF Turtle Syntax

```
:Parent rdf:type owl:Class ;

        owl:equivalentClass [ rdf:type owl:Restriction ;
                              owl:onProperty :hasChild ;
                              owl:someValuesFrom owl:Thing
                            ] .
```

OWL/XML Syntax

```
<EquivalentClasses>
    <Class IRI="#Parent"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#hasChild"/>
        <Class abbreviatedIRI="owl:Thing"/>
    </ObjectSomeValuesFrom>
</EquivalentClasses>
```

Manchester OWL Syntax

```
Class: Parent
    EquivalentTo:
        hasChild some owl:Thing
```

Manchester OWL Syntax

DL Syntax	Manchester Syntax	Example
$C \sqcap D$	C and D	Human and Male
$C \sqcup D$	C or D	Male or Female
$\neg C$	not C	not Male
$\exists R.C$	R some C	hasChild some Human
$\forall R.C$	R only C	hasChild only Human
$(\geq nR.C)$	R min n C	hasChild min 1 Human
$(\leq nR.C)$	R max n C	hasChild max 1 Human
$(= nR.C)$	R exactly n C	hasChild exactly 1 Human
$\{a\} \sqcup \{b\} \sqcup ...$	{a b ...}	{John Robert Mary}

OWL classes and namespaces

- $\perp$ is `owl:Nothing`
- $\top$ is `owl:Thing`
- `owl:` is a *namespace* (`http://www.w3.org/2002/07/owl#`)
- `owl:Thing` expands to `http://www.w3.org/2002/07/owl#Thing` (a class IRI)
- all OWL entities (ontologies, classes, properties, instances) are referred to by an IRI
- namespaces define a common (IRI-)prefix, e.g.,
  - `rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#`
  - `rdfs: http://www.w3.org/2000/01/rdf-schema#`
- can define own namespaces:

```
Namespace: mynamespace <http://www.kaust.edu.sa#>
Class: mynamespace:Student # http://www.kaust.edu.sa#Student
```

Object properties

- Object property characteristics:
  - transitive
  - symmetric, asymmetric
  - reflexive, irreflexive
  - functional, inverse functional
  - inverse of
- Domain and range

Annotation properties

- OWL entities (classes, properties, axioms, ontologies, etc.) can have *annotations*
- outside of OWL semantics (unless for OWL Full)
- useful to add labels, synonyms, explanation, (textual) definitions, authoring information, versions, etc.
- predefined: `rdfs:label`, `owl:versionInfo`, `rdfs:comment`, `rdfs:seeAlso`, `rdfs:isDefinedBy`
- Dublin Core

OWL Reasoning

- Classification: compute the most specific sub- and super-classes for each named class in an OWL ontology
- Subsumption: find all sub-, super- or equivalent classes of an OWL class description
- Consistency: find contradictions in OWL knowledge base
- Instantiation: is a and instance of C?

Complexity of reasoning in OWL

- OWL 2 ($\mathcal{SROIQ}$) is 2NEXPTIME-complete
- OWL (1) ($\mathcal{SHOIN}$) is NEXPTIME-complete
- OWL Lite ($\mathcal{SHIF}$) is EXPTIME-complete

OWL profiles

- OWL 2 EL: PTIME-complete
- OWL 2 RL: PTIME-complete
- OWL 2 QL: AC^0 w.r.t. data size

OWL 2 EL

- Class axioms:
  - subclass, equivalent class, disjoint class
- Object property axioms:
  - domain and range restrictions, property inclusion, property chains, property equivalence, transitive and reflexive properties
- Class descriptions:
  - intersection, existential quantification, enumerations to a single individual
- Assertions: all

Why OWL?

- OWL exploits 20+ years of research on Description Logic
- well-defined semantics
- complexity and decidability well understood
- known algorithms
- scalability demonstrated in practise

Why OWL?

Major benefit is the large number of tools and infrastructure:

- Editors: Protege, WebProtege
- Reasoners: HermiT, Pellet, FaCT++, ELK, KAON2, RACER,...
- Explanation, justification
- Modularization
- APIs (esp. the OWL API)

OWL vs Databases

Database	OWL Ontology
Closed World Assumption	Open World Assumption
Unique Name Assumption	No UNA
Schema constraints data structure	Axioms behave like inference rules

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: Is Draco a friend of Harry Potter?

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: Is Draco a friend of Harry Potter?

- DB: No
- OWL: Don't know

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: How many friends has Harry Potter?

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: How many friends has Harry Potter?

- DB: 2
- OWL: At least 1

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig
- RonWeasley $\neq$ HermioneGranger
- HarryPotter: hasFriend only {HerminoeGranger RonWeasley}

Query: How many friends has Harry Potter?

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig
- RonWeasley $\neq$ HermioneGranger
- HarryPotter: hasFriend only {HerminoeGranger RonWeasley}

Query: How many friends has Harry Potter?

- DB: 2
- OWL: 2

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard

Adding new facts:

- Dumbledore: Wizard
- Fawkes: Phoenix
- Fawkes isPetOf DumbleDore

- DB: Update rejects, constrain violation
- OWL: infer that Dumbledore is Human; infer that Dumbledore is a Wizard

Ontology-based information systems

Ontology like DB schema, instances like data

Advantages:

- Relatively easy to maintain and update schema
- Query answers reflect both schema and data
- Can deal with incomplete information
- Answer intensional and extensional queries

Disadvantages:

- Semantic can seem counter-intuitive (OWA, UNA)
- Query answering (logical entailment) much more difficult

Some examples

- Are cyclin dependent kinases *functionally* more similar to lipid kinases or to riboflavin kinases? How about *phenotypically*?

Some examples

- Are cyclin dependent kinases *functionally* more similar to lipid kinases or to riboflavin kinases? How about *phenotypically*?
- Which protein in the *mouse* is functionally most similar to the zebrafish *gustducin* protein?

Some examples

- Are cyclin dependent kinases *functionally* more similar to lipid kinases or to riboflavin kinases? How about *phenotypically*?
- Which protein in the *mouse* is functionally most similar to the zebrafish *gustducin* protein?
- Which mouse knockout resembles *Bardet-Biedl Syndrome 8*?

Some examples

- Are cyclin dependent kinases *functionally* more similar to lipid kinases or to riboflavin kinases? How about *phenotypically*?
- Which protein in the *mouse* is functionally most similar to the zebrafish *gustducin* protein?
- Which mouse knockout resembles *Bardet-Biedl Syndrome 8*?
- Are there mouse knockouts that resemble the side effects of diclofenac?

Some examples

- Are cyclin dependent kinases *functionally* more similar to lipid kinases or to riboflavin kinases? How about *phenotypically*?
- Which protein in the *mouse* is functionally most similar to the zebrafish *gustducin* protein?
- Which mouse knockout resembles *Bardet-Biedl Syndrome 8*?
- Are there mouse knockouts that resemble the side effects of diclofenac?
- Which genetic disease produces similar symptoms to ebola?

Some examples

- Are cyclin dependent kinases *functionally* more similar to lipid kinases or to riboflavin kinases? How about *phenotypically*?
- Which protein in the *mouse* is functionally most similar to the zebrafish *gustducin* protein?
- Which mouse knockout resembles *Bardet-Biedl Syndrome 8*?
- Are there mouse knockouts that resemble the side effects of diclofenac?
- Which genetic disease produces similar symptoms to ebola?
- Does functional similarity correlate with phenotypic similarity?

Ontologies (recap)

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*

Ontologies (recap)

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*
- *instances* of classes are individuals satisfying the classes' intension
  - my arm, the influenza I had last year, one ethanol molecule, etc.

Ontologies (recap)

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*
- *instances* of classes are individuals satisfying the classes' intension
  - my arm, the influenza I had last year, one ethanol molecule, etc.
- *relations* between instances arise from interactions, configurations, etc., of individuals
  - my arm is **part of** me, the **duration of** my influenza was 10 days

Ontologies (recap)

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*
- *instances* of classes are individuals satisfying the classes' intension
  - my arm, the influenza I had last year, one ethanol molecule, etc.
- *relations* between instances arise from interactions, configurations, etc., of individuals
  - my arm is **part of** me, the **duration of** my influenza was 10 days
- *axioms* specify the conditions that instances of a class must satisfy
  - every instance of *Hand* is a **part of** an instance of *Arm*

Ontologies and graphs

- semantic similarity measures can be graph-based, feature-based, or model-based
- we may need to generate graphs from ontologies
  - *is-a* relations are easy
  - how about *part-of*, *regulates*, *precedes*, etc.?
- relational patterns are implicit in OWL axioms
  - in first order logic
  - needs to translate them into OWL
  - defined in OBO Relation Ontology

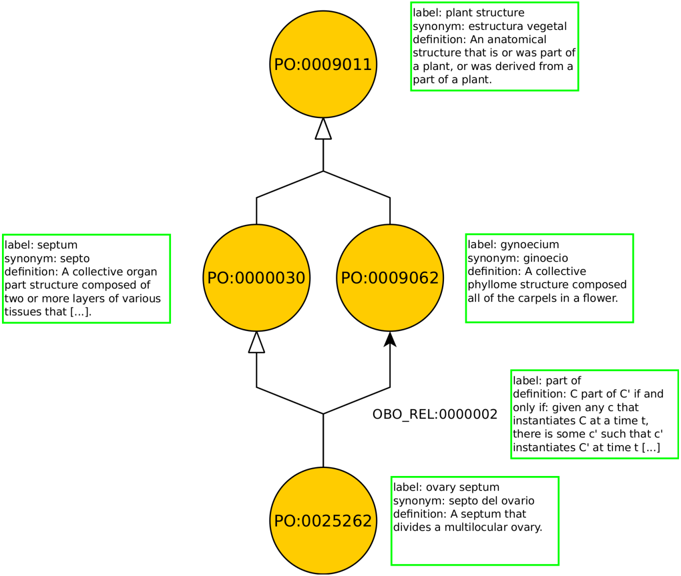
Relations as patterns

- `X SubClassOf: Y`: $X \xrightarrow{\text{is-a}} Y$
- `X SubClassOf: part-of some Y`: $X \xrightarrow{\text{part-of}} Y$
- `X SubClassOf: regulates some Y`: $X \xrightarrow{\text{regulates}} Y$
- `X DisjointWith: Y`: $X \xleftrightarrow{\text{disjoint}} Y$
- `X EquivalentTo: Y`: $X \overset{\equiv}{\leftrightarrow} Y, \{X, Y\}$

Relations as patterns

- OBO Relation Ontology (RO):
  - `https://github.com/oborel/obo-relations`
- Basic Formal Ontology (BFO):
  - provides top-level classes
    - Continuant, Process, Function, Material object, etc.
  - used for some OBO Foundry ontologies
- RO and BFO provide a top-level system of classes and relations shared across many biomedical ontologies
  - even GO, although somewhat hidden!

Relations as patterns



Relations as patterns: Quiz

- *classes*
  a. represent kinds of things in the world
  b. represent ideas in people's heads
  c. represent words

Relations as patterns: Quiz

- *classes*
  - a represent kinds of things in the world
  - b represent ideas in people's heads
  - c represent words
- *instances* of a class are individuals that
  - a satisfy the class intension
  - b satisfy the axioms used to specify the class
  - c are always concrete, material entities

Relations as patterns: Quiz

- *classes*
  a represent kinds of things in the world
  b represent ideas in people's heads
  c represent words
- *instances* of a class are individuals that
  a satisfy the class intension
  b satisfy the axioms used to specify the class
  c are always concrete, material entities
- *relations between classes* are
  a interactions between instances of the class
  b abbreviations of axioms that constrain two classes
  c relations between ideas people have about the classes

Relations as patterns: Quiz

- *classes*
  a represent kinds of things in the world
  b represent ideas in people's heads
  c represent words
- *instances* of a class are individuals that
  a satisfy the class intension
  b satisfy the axioms used to specify the class
  c are always concrete, material entities
- *relations between classes* are
  a interactions between instances of the class
  b abbreviations of axioms that constrain two classes
  c relations between ideas people have about the classes
- *axioms* are
  a specification of conditions that instances of classes must satisfy
  b rules that can be executed to produce new knowledge
  c statements that are considered to be true in a domain of knowledge

How to measure similarity?

- semantic similarity measures similarity between classes
- semantic similarity measures similarity between instances of classes
- semantic similarity measures similarity between entities annotated with classes
- $\Rightarrow$ reduce all of this to similarity between classes

How to measure similarity?

What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function *sim* is:

How to measure similarity?

What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function sim is:

- non-negative: $sim(x, y) \geq 0$ for all x, y

How to measure similarity?

What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function sim is:

- non-negative: $sim(x, y) \geq 0$ for all x, y
- symmetric: $sim(x, y) = sim(y, x)$

How to measure similarity?

What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function *sim* is:

- non-negative: $sim(x, y) \geq 0$ for all x, y
- symmetric: $sim(x, y) = sim(y, x)$
- reflexive: $sim(x, x) = max_D$

How to measure similarity?

What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function sim is:

- non-negative: $sim(x, y) \geq 0$ for all x, y
- symmetric: $sim(x, y) = sim(y, x)$
- reflexive: $sim(x, x) = max_D$
  - weaker form: $sim(x, x) > sim(x, y)$ for all $x \neq y$

How to measure similarity?

What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function sim is:

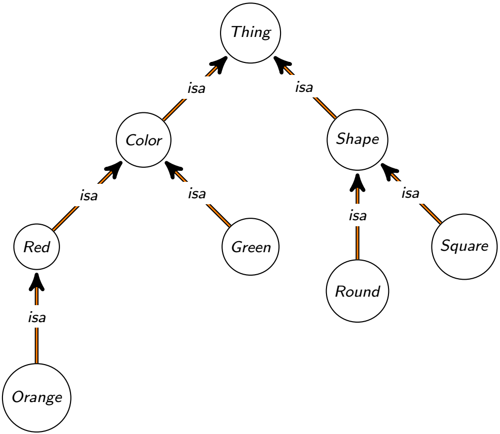
- non-negative: $sim(x, y) \geq 0$ for all x, y
- symmetric: $sim(x, y) = sim(y, x)$
- reflexive: $sim(x, x) = max_D$
  - weaker form: $sim(x, x) > sim(x, y)$ for all $x \neq y$
- $sim(x, x) > sim(x, y)$ for $x \neq y$

How to measure similarity?

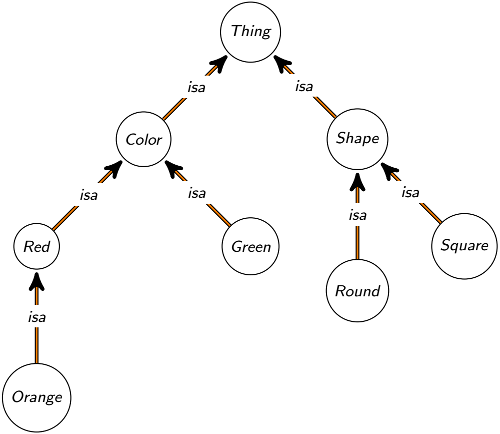
What properties do we want in a similarity measure?
A function $sim : D \times D$ is a similarity on D if, for all $x, y \in D$, the function sim is:

- non-negative: $sim(x, y) \geq 0$ for all x, y
- symmetric: $sim(x, y) = sim(y, x)$
- reflexive: $sim(x, x) = max_D$
  - weaker form: $sim(x, x) > sim(x, y)$ for all $x \neq y$
- $sim(x, x) > sim(x, y)$ for $x \neq y$
- sim is a *normalized* similarity measure if it has values in $[0, 1]$

How to measure similarity?

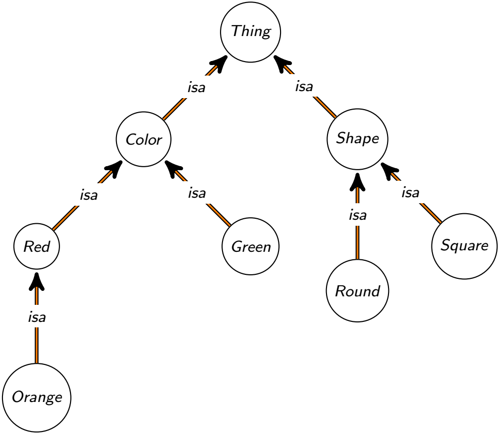


How to measure similarity?



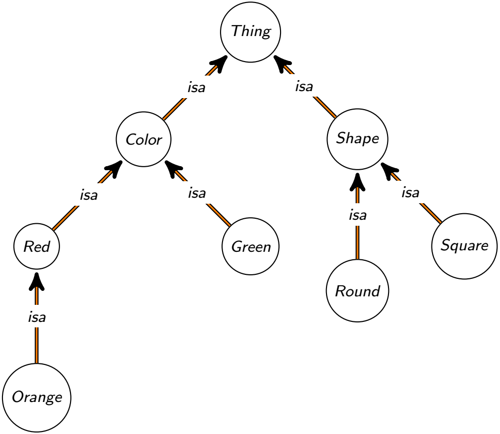
- distance on shortest path (Rada *et al.*, 1989)

How to measure similarity?



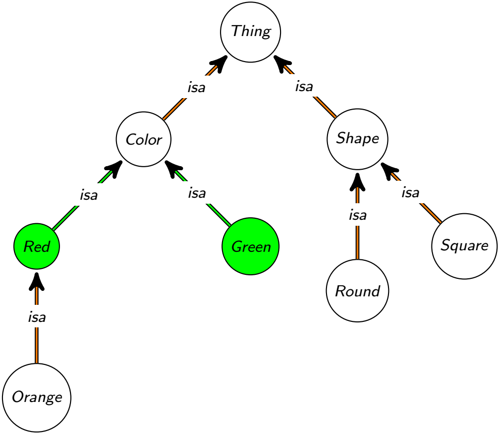
- distance on shortest path (Rada *et al.*, 1989)
- $dist_{Rada}(u, v) = sp(u, isa, v)$

How to measure similarity?



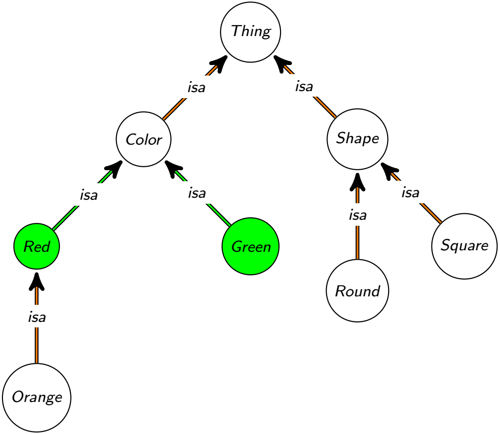
- distance on shortest path (Rada *et al.*, 1989)
- $dist_{Rada}(u, v) = sp(u, isa, v)$
- $sim_{Rada}(u, v) = \frac{1}{dist_{Rada}(u,v)+1}$

How to measure similarity?



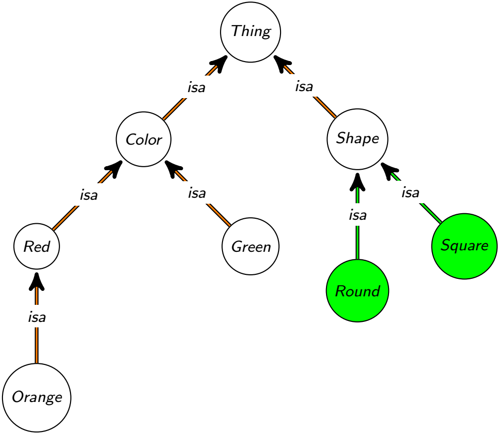
- distance on shortest path

How to measure similarity?



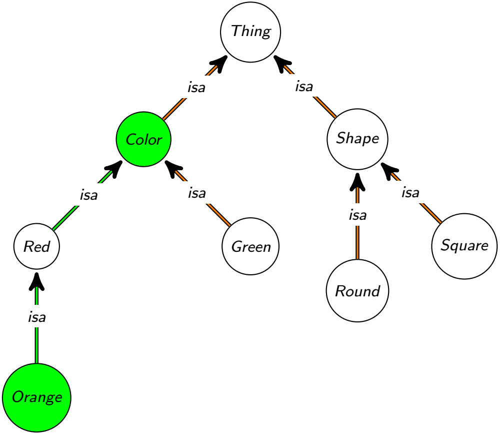
- distance on shortest path
- distance(green, red) $= 2$
- $sim_{Rada}(green, red) = \frac{1}{3}$

How to measure similarity?



- distance on shortest path
- distance(square, round) = 2
- $sim_{Rada}(square, round) = \frac{1}{3}$

How to measure similarity?



- distance on shortest path
- distance(orange, color) = 2
- $sim_{Rada}(orange, color) = \frac{1}{3}$

How to measure similarity?

- shortest path is not always intuitive

How to measure similarity?

- shortest path is not always intuitive
- we need a way to determine *specificity* of a class
  - number of ancestors
  - number of children
  - information content

How to measure similarity?

- shortest path is not always intuitive
- we need a way to determine *specificity* of a class
  - number of ancestors
  - number of children
  - information content
- *density* of a branch in the ontology
  - number of siblings
  - information content

How to measure similarity?

- shortest path is not always intuitive
- we need a way to determine *specificity* of a class
    - number of ancestors
    - number of children
    - information content
- *density* of a branch in the ontology
    - number of siblings
    - information content
- account for different edge types
    - non-uniform edge weighting

How to measure similarity

- term specificity measure $\sigma : C \mapsto \mathbb{R}$:
  - $x \sqsubseteq y \rightarrow \sigma(x) \geq \sigma(y)$

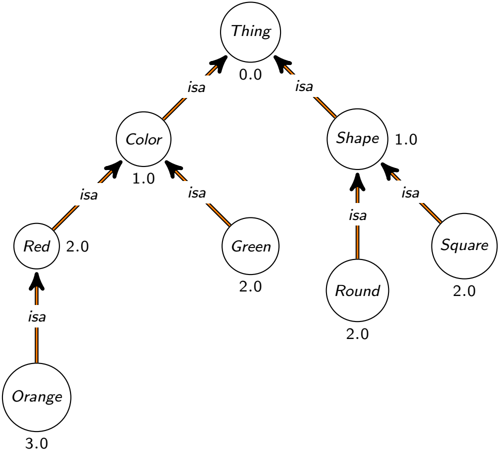
How to measure similarity

- term specificity measure $\sigma : C \mapsto \mathbb{R}$:
  - $x \sqsubseteq y \rightarrow \sigma(x) \geq \sigma(y)$
- intrinsic:
  - $\sigma(x) = f(depth(x))$
  - $\sigma(x) = f(A(x))$ (for ancestors $A(x)$)
  - $\sigma(x) = f(D(x))$ (for descendants $D(x)$)
  - many more, e.g., Zhou et al.:
    $\sigma(x) = k \cdot \left(1 - \frac{\log |D(x)|}{\log |C|}\right) + (1-k)\frac{\log depth(x)}{\log depth(G_T)}$

How to measure similarity

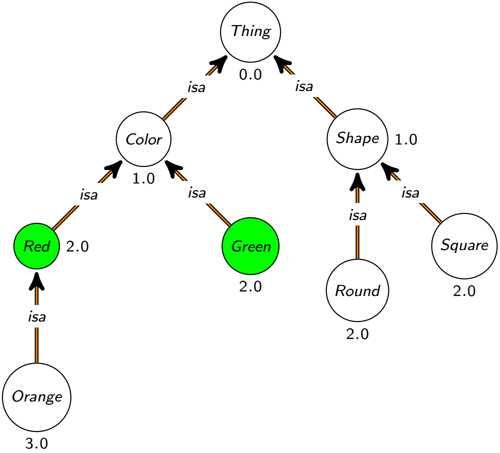
- term specificity measure $\sigma : C \mapsto \mathbb{R}$:
  - $x \sqsubseteq y \rightarrow \sigma(x) \geq \sigma(y)$
- intrinsic:
  - $\sigma(x) = f(depth(x))$
  - $\sigma(x) = f(A(x))$ (for ancestors $A(x)$)
  - $\sigma(x) = f(D(x))$ (for descendants $D(x)$)
  - many more, e.g., Zhou et al.:
    $\sigma(x) = k \cdot \left(1 - \frac{\log |D(x)|}{\log |C|}\right) + (1-k)\frac{\log depth(x)}{\log depth(G_T)}$
- extrinsic:
  - $\sigma(x)$ defined as a function of instances (or annotations) I
    - note: the number of instances monotonically decreases with increasing depth in taxonomies
  - Resnik 1995: $eIC_{Resnik}(x) = -\log p(x)$ (with $p(x) = \frac{|I(x)|}{|I|}$)
    - in biology, one of the most popular specificity measure when annotations are present

How to measure similarity?



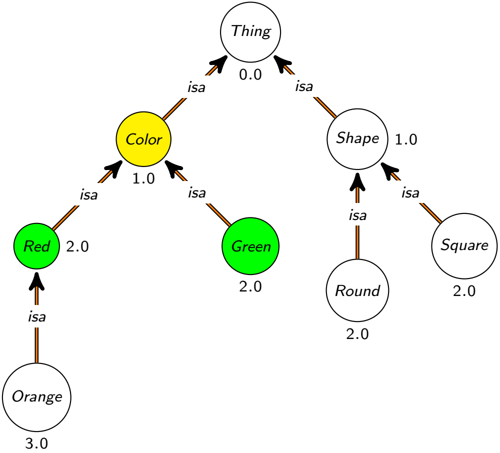
- Resnik 1995: similarity between x and y is the information content of the *most informative common ancestor*

How to measure similarity?



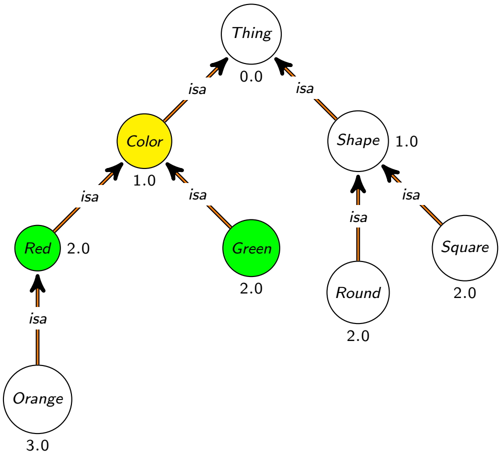
- Resnik 1995: similarity between x and y is the information content of the *most informative common ancestor*

How to measure similarity?



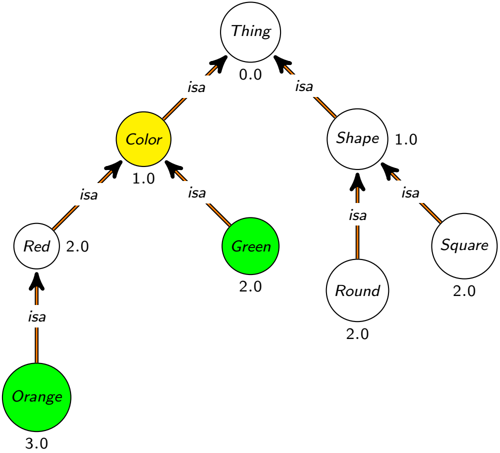
- Resnik 1995: similarity between x and y is the information content of the *most informative common ancestor*

How to measure similarity?



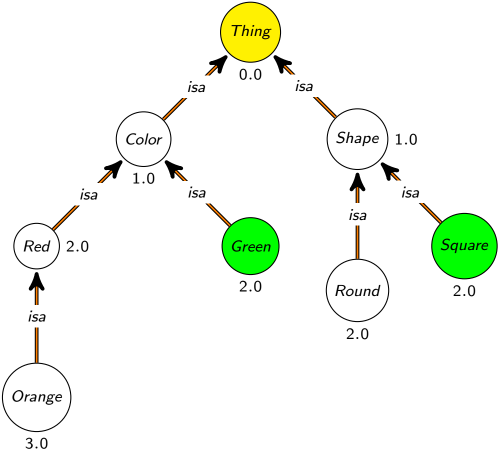
- Resnik 1995: similarity between x and y is the information content of the *most informative common ancestor*
- $sim_{Resnik}(Green, Red) = 1.0$

How to measure similarity?



- Resnik 1995: similarity between x and y is the information content of the *most informative common ancestor*
- $sim_{Resnik}(Green, Orange) = 1.0$

How to measure similarity?

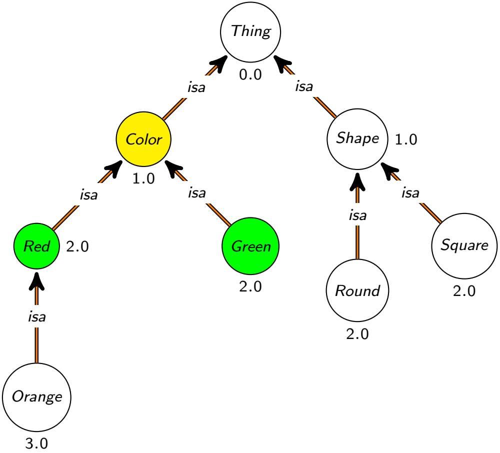


- Resnik 1995: similarity between x and y is the information content of the *most informative common ancestor*
- $sim_{Resnik}(Square, Orange)$ 0.0

How to measure similarity?

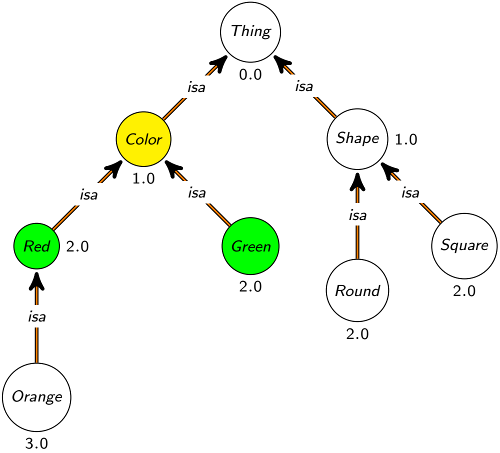
- (Red, Green) and (Orange, Green) have the same similarity
- need to incorporate the specificity of the compared classes

How to measure similarity?



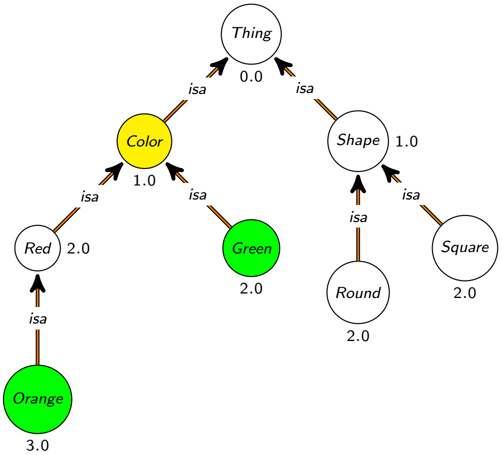
- Lin 1998: $sim_{Lin}(x, y) = \frac{2 \cdot IC(MICA(x,y))}{IC(x)+IC(y)}$

How to measure similarity?



- Lin 1998: $sim_{Lin}(x, y) = \frac{2 \cdot IC(MICA(x,y))}{IC(x)+IC(y)}$
- $sim_{Lin}(Green, Red) = 0.5$

How to measure similarity?



- Lin 1998: $sim_{Lin}(x, y) = \frac{2 \cdot IC(MICA(x,y))}{IC(x)+IC(y)}$
- $sim_{Lin}(Green, Orange) = 0.4$

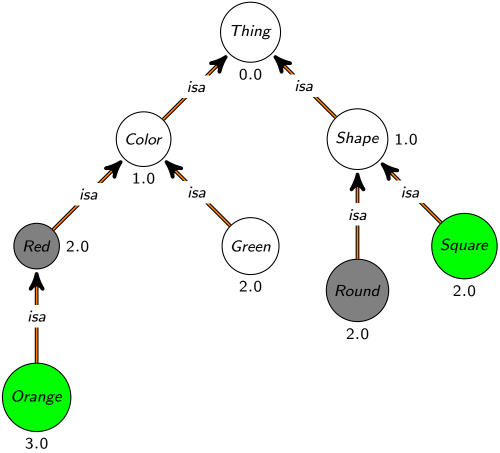
How to measure similarity?

- many(!) others:
  - Jiang & Conrath 1997
  - Mazandu & Mulder 2013
  - Schlicker et al. 2009
  - ...

How to measure similarity?

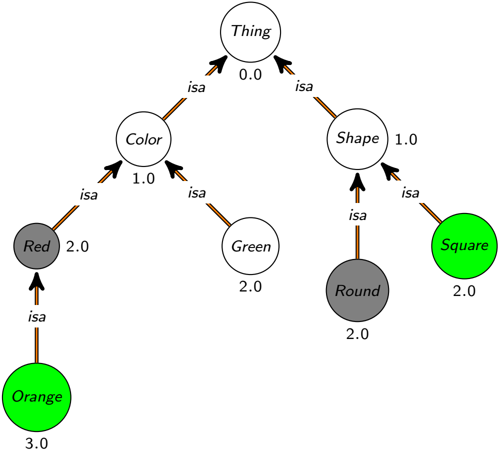
- we only looked at comparing pairs of classes
- mostly, we want to compare *sets* of classes
  - set of GO annotations
  - set of signs and symptoms
  - set of phenotypes
- two approaches:
  - compare each class individually, then merge
  - directly set-based similarity measures

How to measure similarity?



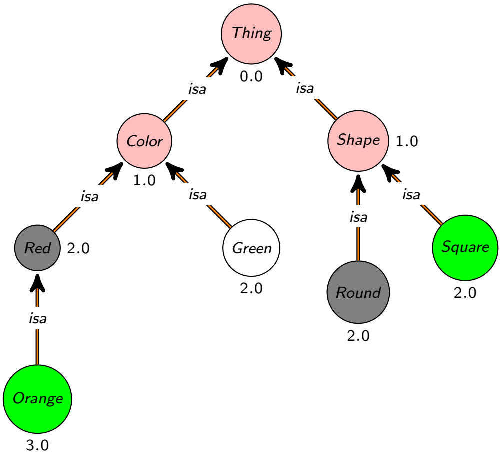
- similarity between a square-and-orange thing and a round-and-red thing

How to measure similarity?



- similarity between a square-and-orange thing and a round-and-red thing
- Pesquita et al., 2007: $simGIC(X, Y) = \frac{\sum_{c \in A(X) \cap A(Y)} IC(c)}{\sum_{c \in A(X) \cup A(Y)} IC(c)}$

How to measure similarity?



- similarity between a square-and-orange thing and a round-and-red thing
- Pesquita et al., 2007: $simGIC(X, Y) = \frac{\sum_{c \in A(X) \cap A(Y)} IC(c)}{\sum_{c \in A(X) \cup A(Y)} IC(c)}$
- $simGIC(so, rr) = \frac{2}{11}$

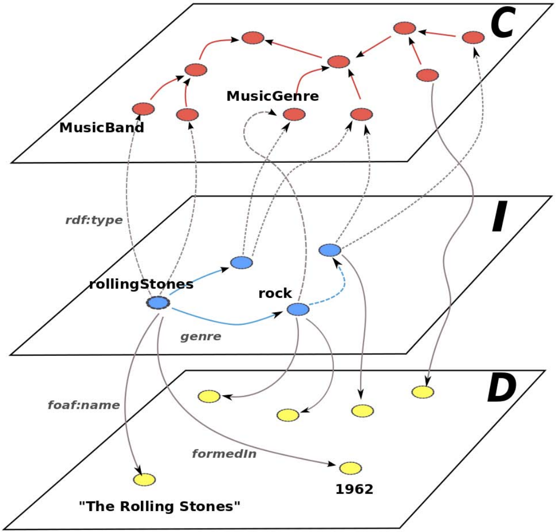
How to measure similarity?

- alternatively: use different merging strategies
- common: average, maximum, **best-matching average**
  - Average: $sim_A(X, Y) = \frac{\sum_{x \in X} \sum_{y \in Y} sim(x,y)}{|X| \times |Y|}$
  - Max average: $sim_{MA}(X, Y) = \frac{1}{|X|} \sum_{x \in X} \max_{y \in Y} sim(x, y)$
  - Best match average: $sim_{BMA}(X, Y) = \frac{sim_{MA}(X,Y) + sim_{MA}(Y,X)}{2}$

How to measure similarity?

- Semantic Measures Library:
  - comprehensive Java library
  - `http://www.semantic-measures-library.org/`
- R packages: GOSim, GOSemSim, HPOSim, LSAfun, ontologySimilarity,...
- Python: sematch, fastsemsim (GO only)

How to measure similarity?



From Harispe et al., Semantic Similarity From Natural Language And Ontology Analysis, 2015.

How to measure similarity?

- Shortest Path
  - applicable to arbitrary knowledge graphs
  - does not capture similarity well over all edge types, e.g., *disjointWith*, *differentFrom*, *opposite-of*, etc.
- Random Walk
  - with or without restart
  - iterated
  - does not consider edge labels $\Rightarrow$ captures only adjacency of nodes
  - scores whole graph with *probability* of being in a state
  - can take multiple seed nodes
    - widely used to find disease genes

How to measure similarity?

- feature learning on knowledge graph

How to measure similarity?

- feature learning on knowledge graph
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)

How to measure similarity?

- feature learning on knowledge graph
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)
  - with support for reasoning over bio-ontologies: `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`
  - Onto2Vec: `https://github.com/bio-ontology-research-group/onto2vec/`

How to measure similarity?

- feature learning on knowledge graph
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)
  - with support for reasoning over bio-ontologies: `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`
  - Onto2Vec: `https://github.com/bio-ontology-research-group/onto2vec/`
- Translational knowledge graph embeddings: TransE, TransR, TransE, HolE, etc.
  - analogy-based
  - `https://github.com/thunlp/KB2E`

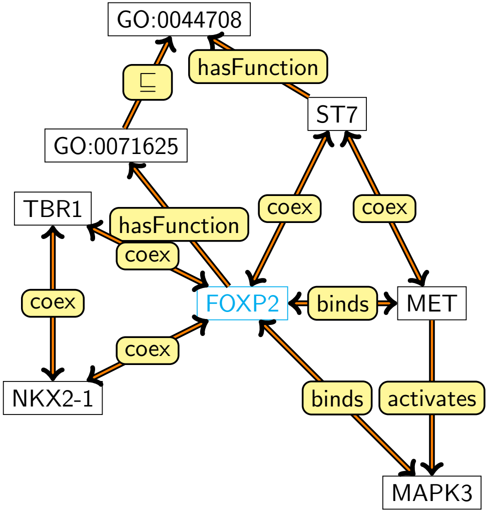
How to measure similarity?

- feature learning on knowledge graph
- e.g., iterated, edge-labeled random walk
  - walks form *sentences*
  - sentences form a *corpus*
  - feature learning on corpus through Word2Vec (or factorization of co-occurrence matrix)
  - with support for reasoning over bio-ontologies: `https://github.com/bio-ontology-research-group/walking-rdf-and-owl`
  - Onto2Vec: `https://github.com/bio-ontology-research-group/onto2vec/`
- Translational knowledge graph embeddings: TransE, TransR, TransE, HolE, etc.
  - analogy-based
  - `https://github.com/thunlp/KB2E`
- generates (dense) feature vectors for nodes (classes, instances) and relations

Knowledge graph embeddings

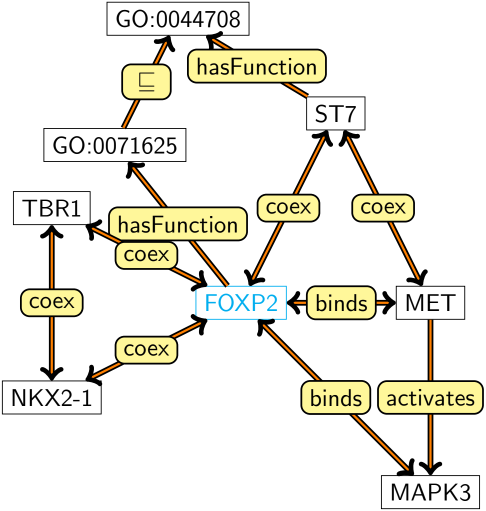
Definition
Let $KG = (V, E, L; \vdash)$ be a knowledge graph with a set of vertices V, a set of edges $E \subseteq V \times V$, a label function $L : V \cup E \mapsto Lab$ that assigns labels from a label set Lab to vertices and edges, and an inference relation $\vdash$. A knowledge graph embedding is a function $f_\eta : KG \mapsto \mathbf{R}^n$ (subject to certain constraints).

Neuro-symbolic feature learning

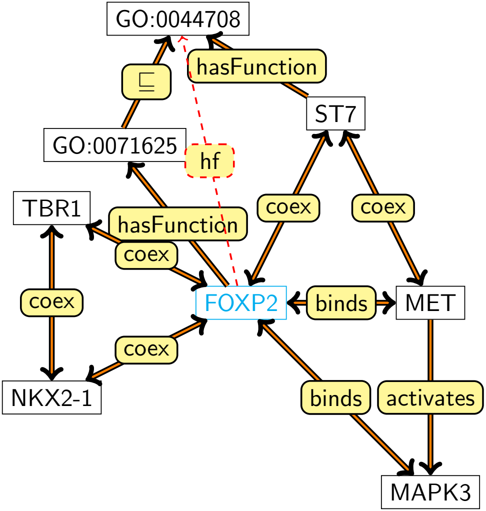


- task: predict if FOXP2 is involved in disease D
- task: what chemicals could (directly or indirectly) affect FOXP2's function?
- which features are relevant?

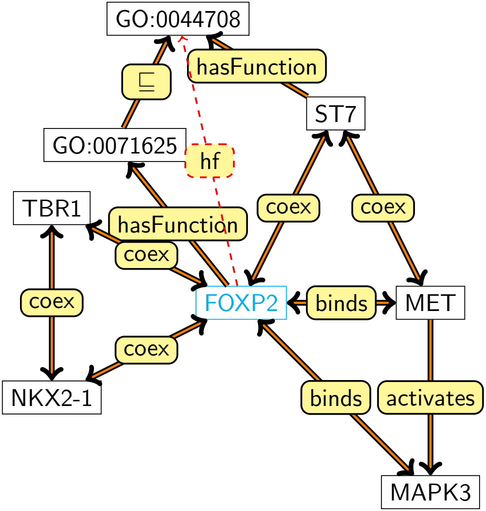
Neuro-symbolic feature learning



Neuro-symbolic feature learning

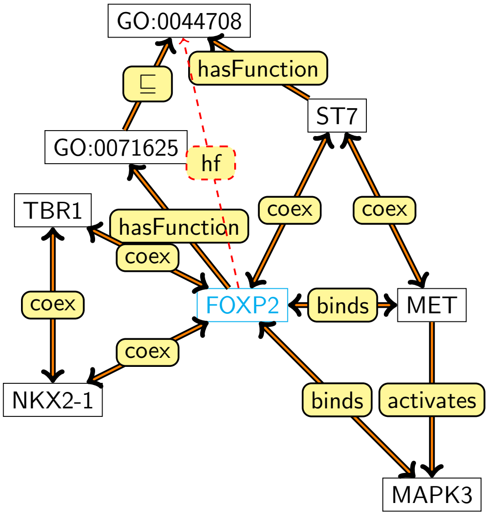


Neuro-symbolic feature learning



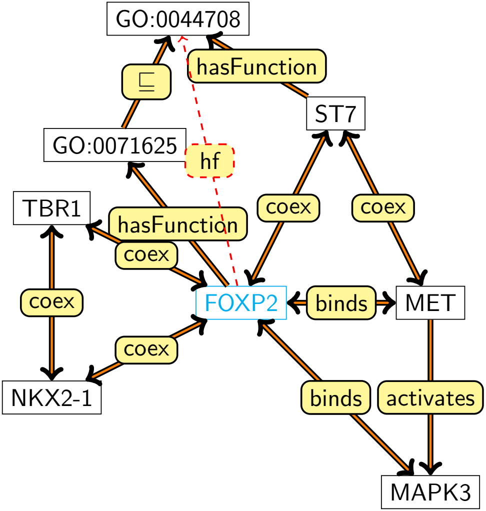
- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708

Neuro-symbolic feature learning



- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708

Neuro-symbolic feature learning

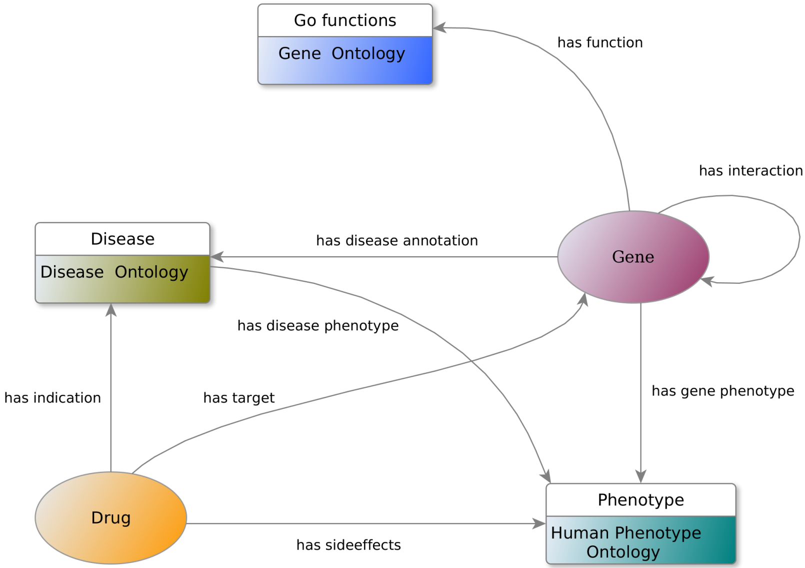


- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708
- :FOXP2 :coex :TBR1 :coex :NKX2-1 :coex :TBR1 :coex ...

Neuro-symbolic feature learning

- skip-gram model learns representation/features for each node
  - Word2Vec model, given a word predicts context
  - use local and non-local information
- automated reasoning deductively closes the knowledge graph
  - making this a neuro-symbolic model
- useful for edge prediction, similarity, clustering, as feature vectors
  - edge prediction: analogy, classifier (e.g., SVM)

Neuro-symbolic feature learning



Neuro-symbolic feature learning

Object property	Source type	Target type	Without reasoning		With reasoning	
			F-measure	AUC	F-measure	AUC
has target	Drug	Gene/Protein	0.94	0.97	0.94	0.98
has disease annotation	Gene/Protein	Disease	0.89	0.95	0.89	0.95
has side-effect*	Drug	Phenotype	0.86	0.93	0.87	0.94
has interaction	Gene/Protein	Gene/Protein	0.82	0.88	0.82	0.88
has function*	Gene/Protein	Function	0.85	0.95	0.83	0.91
has gene phenotype*	Gene/Protein	Phenotype	0.84	0.91	0.82	0.90
has indication	Drug	Disease	0.72	0.79	0.76	0.83
has disease phenotype*	Disease	Phenotype	0.72	0.78	0.70	0.77

Alsharani et al. Neuro-symbolic representation learning on biological knowledge graphs. Bioinformatics, 2017.

Multi-modal feature learning

The forkhead-box P2 (FOXP2) gene polymorphism has been reported to be involved in the susceptibility to schizophrenia; however, few studies have investigated the association between FOXP2 gene polymorphism and clinical symptoms in schizophrenia.

Multi-modal feature learning

> *The forkhead-box P2 (FOXP2) gene polymorphism has been reported to be involved in the susceptibility to schizophrenia; however, few studies have investigated the association between FOXP2 gene polymorphism and clinical symptoms in schizophrenia.*

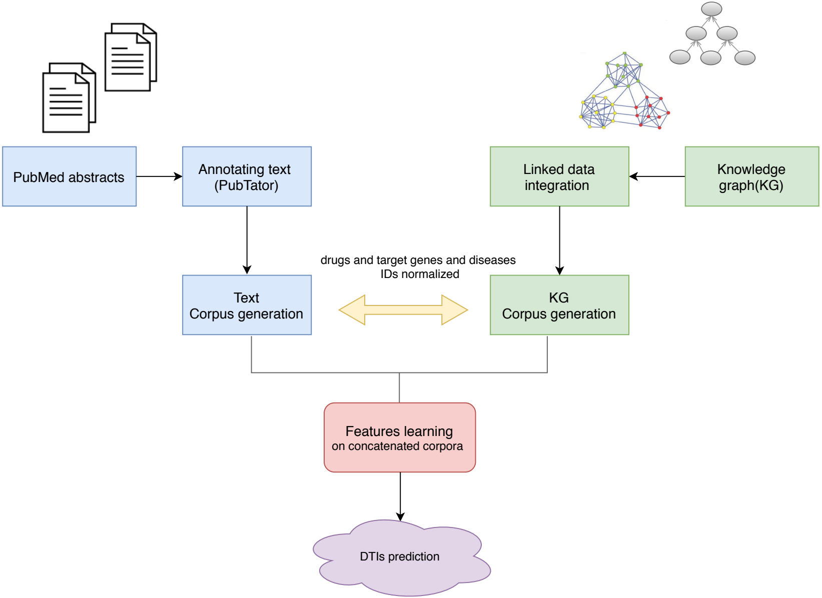
- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708
- :FOXP2 :coex :TBR1 :coex :NKX2-1 :coex :TBR1 :coex ...

Multi-modal feature learning

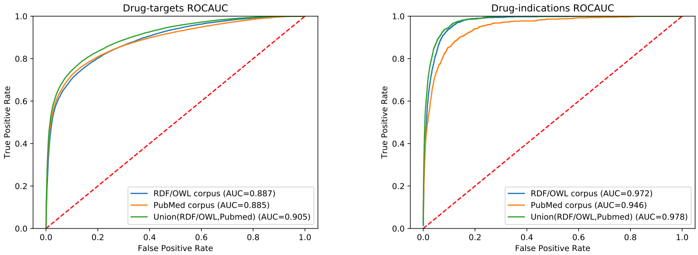
The :FOXP2 gene polymorphism has been reported to be involved in the susceptibility to schizophrenia; however, few studies have investigated the association between :FOXP2 gene polymorphism and clinical symptoms in schizophrenia.

- :FOXP2 :binds :MET :coex :ST7 :hasFunction GO:0044708
- :FOXP2 :hasFunction GO:0071625 subClassOf GO:0044708
- :FOXP2 :coex :TBR1 :coex :NKX2-1 :coex :TBR1 :coex ...

Multi-modal feature learning



Multi-modal feature learning: drug targets and indications



Alshahrani & H. Drug repurposing through multi-modal learning on knowledge graphs. BioRxiv, 2018.

Ontologies: axioms, not graphs!

Overview | Browse | DLQuery | Download

Annotation	Value
label	B cell apoptotic process
definition	Any apoptotic process in a B cell, a lymphocyte of B lineage with the phenotype CD19-positive and capable of B cell mediated immunity.
class	http://purl.obolibrary.org/obo/GO_0001783
ontology	GO-PLUS
Equivalent	apoptotic process and (occurs in some B cell)
SubClassOf	occurs in some B cell, lymphocyte apoptotic process
id	GO:0001783
has_obo_namespace	biological_process

Ontologies: axioms, not graphs!

Gene Ontology:

- `behavior DisjointWith: 'developmental process'`
- `behavior SubclassOf: only-in-taxon some metazoa`
- `'cell proliferation' DisjointWith: in-taxon some fungi`
- `'cell growth' EquivalentTo: growth and ('results in growth of' some cell)`
- ...

Ontology embeddings

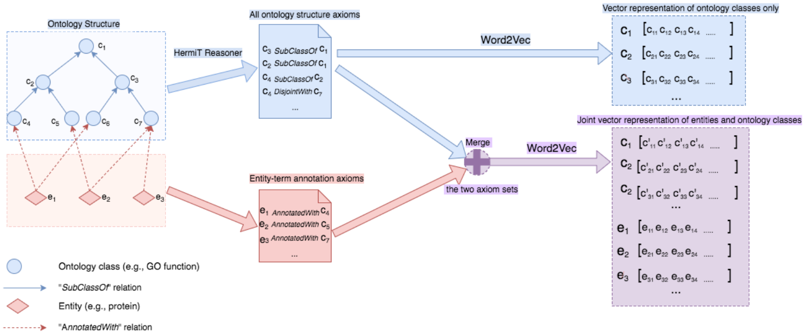
Definition
Let $O = (C, R, I; ax; \vdash)$ be an ontology with a set of classes C, a set of relations R, a set of instances I, a set of axioms ax and an inference relation $\vdash$. An ontology embedding is a function $f_\eta : C \cup R \cup I \mapsto \mathbf{R}^n$ (subject to certain constraints).

Ontology embeddings

Definition
Let $O = (C, R, I; ax; \vdash)$ be an ontology with a set of classes C, a set of relations R, a set of instances I, a set of axioms ax and an inference relation $\vdash$. An ontology embedding is a function $f_\eta : C \cup R \cup I \mapsto \mathbf{R}^n$ (subject to certain constraints).

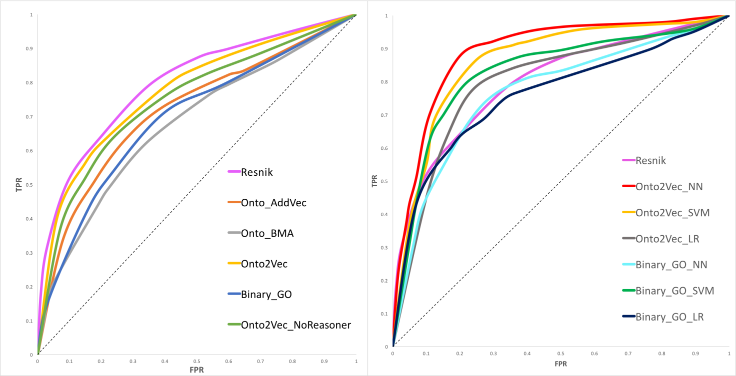
We use co-occurrence within $ax^{\vdash}$ to constrain the embedding function, where the constraints on co-occurrence are formulated using the Word2Vec skipgram model.

Onto2Vec



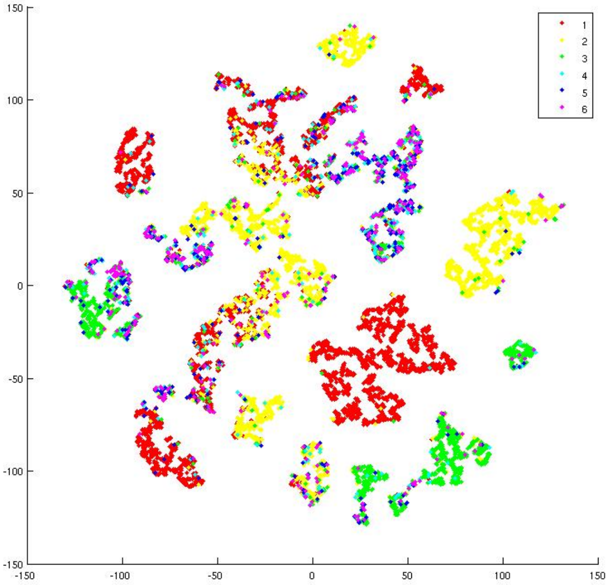
Ontology Structure
HermiT Reasoner
All ontology structure axioms
Word2Vec
Vector representation of ontology classes only
Merge
the two axiom sets
Word2Vec
Joint vector representation of entities and ontology classes
Entity-term annotation axioms
Ontology class (e.g., GO function)
"SubClassOf" relation
Entity (e.g., protein)
"AnnotatedWith" relation

Predicting PPIs: trainable similarity measures

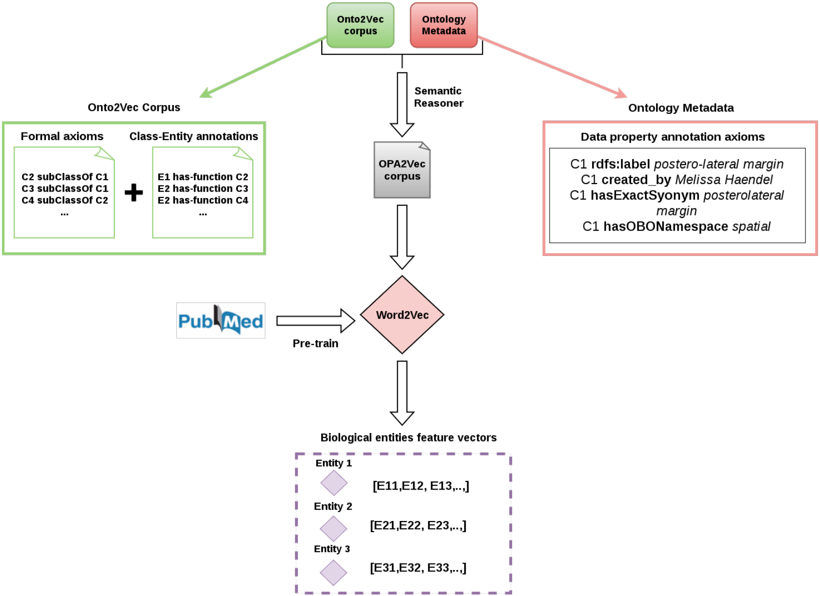


Smaili et al. Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations, Bioinformatics, 2018.

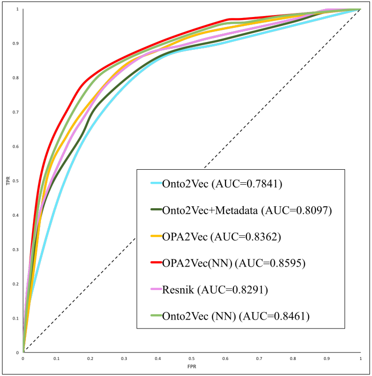
Visualizing embeddings

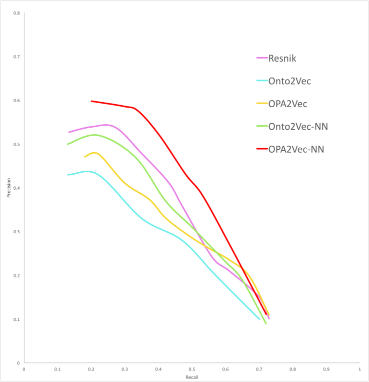


Ontologies Plus Annotations 2 Vec



Phenotype-based prediction of candidate genes





How to measure similarity?

- vector-based similarity measure
- cosine similarity: $sim(X, Y) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \sqrt{\sum_{i=1}^{n} Y_i^2}}$
  - bounded between $[-1, 1]$
- Euclidean distance: $sim(X, Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$
  - not bounded (and rarely used)
- any other kind of function
  - Neural Networks can approximate *any* function (universal approximation theorem)
  - "trainable" semantic similarity measures

How to measure similarity?

- many graph based semantic similarity measures for comparing two classes
- several set-based measures
  - directly set-based
  - merging pair-wise comparison
- most useful when comparing instances/annotations
- other approaches consider relations between instances:
  - path-based
  - random-walk
- very recent: knowledge graph embeddings
  - and any vector-based similarity measure

How to measure similarity?

Recommended reading:

- recommended, comprehensive overview: Sebastian Harispe et al. Semantic Similarity from Natural Language and Ontology Analysis. Morgan & Claypool Publishers, 2015
- Catia Pesquita et al. Semantic Similarity in Biomedical Ontologies. PLoS CB, 2009.
- Maximilian Nickel et al. A Review of Relational Machine Learning for Knowledge Graphs, Proceedings of the IEEE, 2016.

How to measure similarity: Quiz

- How many semantic similarity measures are there?
  - a One (and it is called The Semantic Similarity Measure)
  - b Three (graph-based, set-based, feature-based)
  - c Many (depending on context, many functions can determine similarity)

How to measure similarity: Quiz

- How many semantic similarity measures are there?
  a One (and it is called The Semantic Similarity Measure)
  b Three (graph-based, set-based, feature-based)
  c Many (depending on context, many functions can determine similarity)
- Specificity of an ontology class
  a depends on the number of children and ancestors, and the depth
  b depends on the number of instances (or annotations)
  c can improve similarity estimates significantly

How to measure similarity: Quiz

- How many semantic similarity measures are there?
  a One (and it is called The Semantic Similarity Measure)
  b Three (graph-based, set-based, feature-based)
  c Many (depending on context, many functions can determine similarity)
- Specificity of an ontology class
  a depends on the number of children and ancestors, and the depth
  b depends on the number of instances (or annotations)
  c can improve similarity estimates significantly
- In the presence of (relations between) instances, semantic similarity
  a cannot be computed, it only works with ontologies
  b can be estimated using only class specificity measures
  c can be computed using knowledge graph embeddings

Applications of semantic similarity

- ontologies are used *almost everywhere* in biology
- many applications of semantic similarity:
  - predicting interacting proteins
  - predict candidate genes
    - using the guilt-by-association principle, or without
  - predict drug targets and indications
  - as features in machine learning models

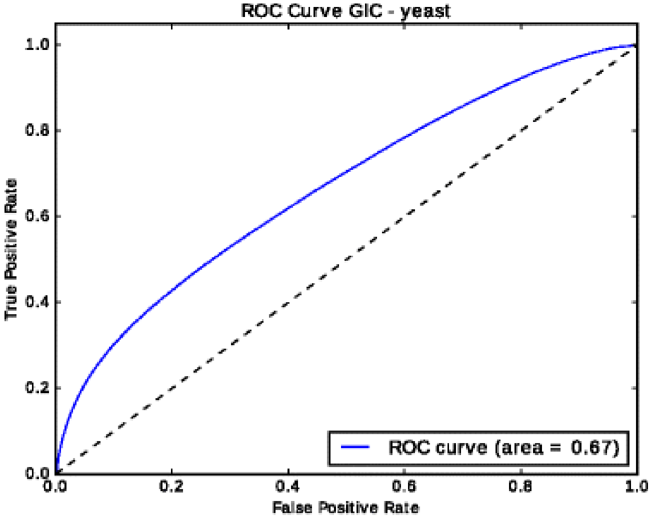
Applications of semantic similarity

Hypothesis

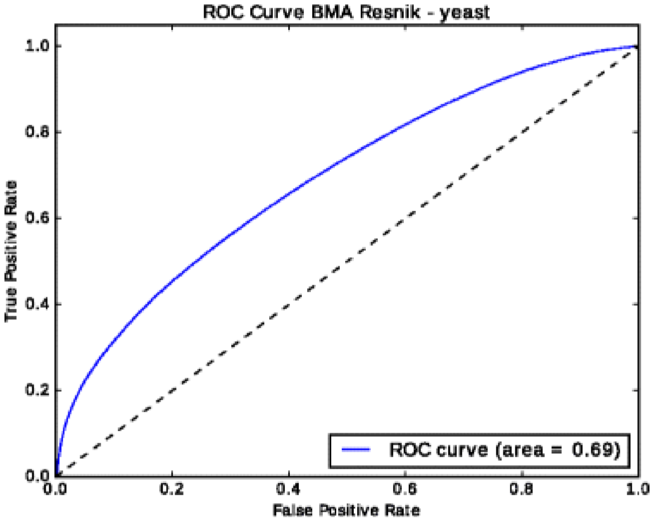
Interacting proteins have similar functions.

- relies on background knowledge about functions (encoded in GO)
- "similarity" can mean:
  - part of the same pathway
  - siblings of a common super-class
  - located in the same location
- set-based comparison of GO functions
  - single GO hierarchy or all?
  - which similarity measure?

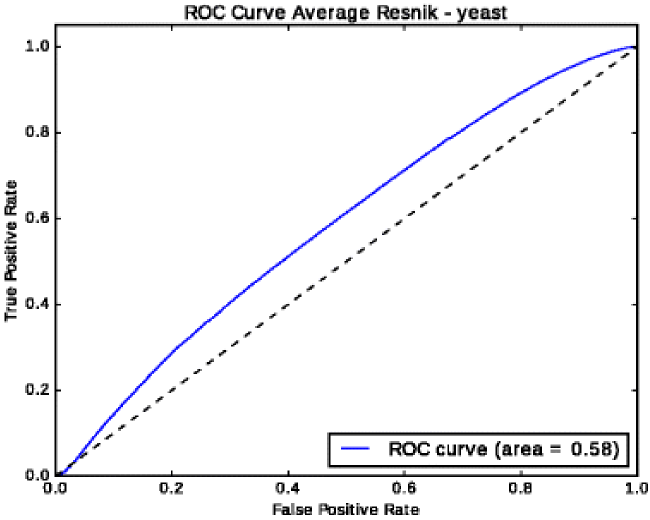
Applications of semantic similarity



Applications of semantic similarity



Applications of semantic similarity



Applications of semantic similarity

- no obvious choice of similarity measure
- depends on application
  - predicting PPIs in different organisms may benefit from a different similarity measure!
- different similarity measures may react differently to biases in data
- needs some testing and experience

Applications of semantic similarity

Recommendations:

- use Resnik's information content measure
- use Resnik's similarity
- use Best Match Average
- use the full ontology
- classify your ontology using an automated reasoner before applying semantic similarity
  - although many ontologies come pre-classified
- $\Rightarrow$ but there are many exceptions
  - similar location $\Rightarrow$ use location subset of GO
  - developmental phenotypes $\Rightarrow$ use developmental branch of phenotype ontology

Onto2Vec and OPA2Vec

Using feature learning to "learn" semantic similarity measures in a data- and application-driven way...

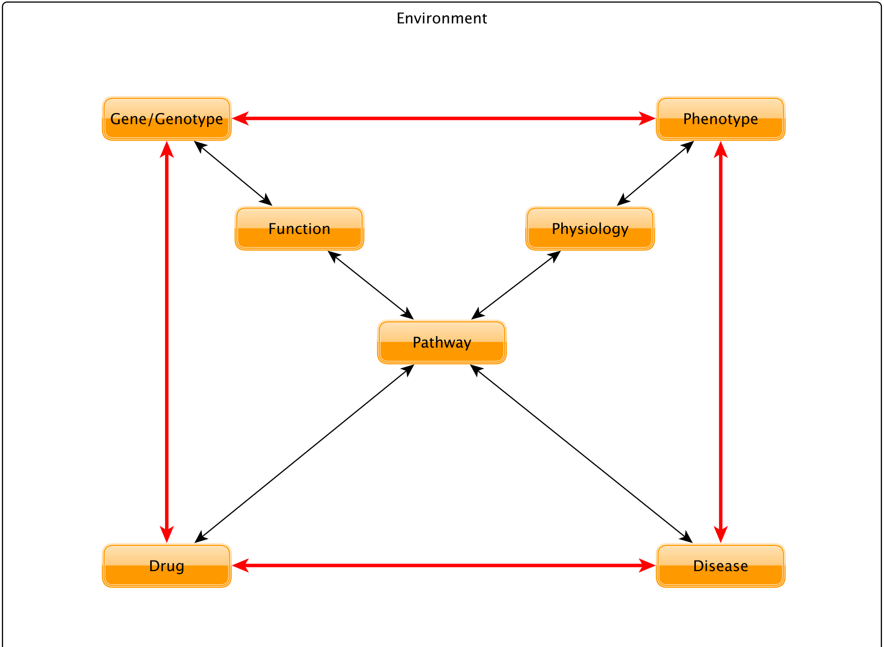
Applications of semantic similarity

- choice of ontology determines the kind of similarity
- functional similarity: Gene Ontology
- anatomical, structural similarity: anatomy ontologies (Uberon, MA, FMA, etc.)
- phenotypic similarity: phenotype ontology (HPO, MP, etc.)
- chemical structural similarity: ChEBI

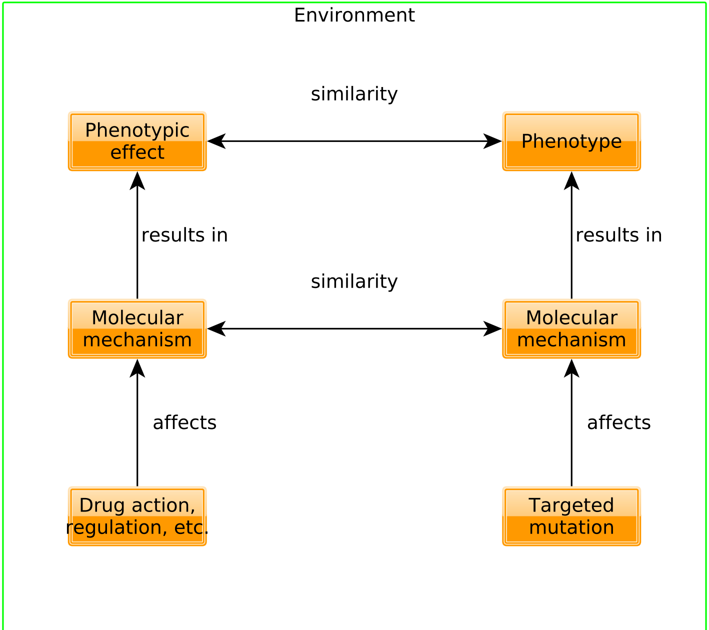
Applications of semantic similarity

- phenotypic similarity used to:
  - diagnosis: similarity between patient phenotypes and disease phenotypes
    - also between patient phenotypes and gene–phenotype associations
    - Phenomizer: `http://compbio.charite.de/phenomizer/`
  - disease modules: similarity between disease and disease
  - clustering/stratification: similarity between patient and patient
  - disease gene discovery: similarity between patient/disease phenotypes and gene–phenotype associations
    - in humans
    - in model organisms
  - drug repurposing: side-effect similarity; similarity between side effect profile and gene–disease associations

Applications of semantic similarity



Applications of semantic similarity



Applications of semantic similarity

- Guilt-by-association:
  - x is associated with y
  - z is similar to x
  - therefore: z may be associated with y
- candidate genes (polygenic disease):
  - FunSimMat: similar function $\Rightarrow$ similar/same disease
  - side effect similarity: similar side effects $\Rightarrow$ similar targets/indications

Applications of semantic similarity

- No guilt-by-association (abduction):
  - x causes a
  - y has b
  - a similar to b
  - therefore: b is caused by x
- candidate genes (monogenic and polygenic disease):
  - Phenomizer: gene x causes phenotypes a; patient y has symptoms b; a is similar to b; therefore: gene x causes the symptoms in b
  - PhenomeNET: similar to Phenomizer but using model organism phenotypes (knockouts)
  - PhenomeDrug: knockout of gene x causes phenotypes a; drug y causes side effects b; a is similar to b; therefore: drug y inhibits x (or: phenotypes b are caused by inhibition of x)
  - needs to compare model organism phenotypes and human phenotypes $\Rightarrow$ ontology alignment/integration/mapping

Applications of semantic similarity

- comparing entities annotated with *different* ontologies/vocabularies of the *same* (or related) domains
  - medical: UMLS, HPO, DO, ORDO, NCIT, ICD, SNOMED CT, MeSH, ...
  - phenotype: HPO, MP, CPO, WBPhenotype, FBCV, MeSH, ...
  - chemical: ChEBI, MeSH, DrOn, RXNorm, DrugBank, ...
- needs mapping, alignment, or integration
  - mapping: given a term t, find corresponding class in ontology O
    - can be 1:1, 1:n, n:1, n:m
    - t can be from ontology, vocabulary, database, or text
    - use O for analysis
  - alignment: given two ontologies or vocabularies O_1 and O_2, find all mappings between classes/terms in O_1 and O_2
    - applicable to ontologies and vocabularies
    - use O_1 or O_2 for analysis
  - integration: given two ontologies O_1 and O_2, combine both ontologies into a single ontology O
    - maintain meaning of classes
    - use O for analysis

Applications of semantic similarity

- lexical mappings: use class labels (and synonyms) to find matches
  - hypertension (`HP:0000822`) and hypertension (`MP:0000231`)
- semantic mappings: use class axioms to find matches
  - pulmonary valve stenosis (`MP:0006182`) and Pulmonic stenosis (`HP:0001642`)
  - both definitions based on constricted (`PATO:0001847`) and pulmonary valve (`UBERON:0002146`)
- hybrid: combine lexical and semantic mappings

Applications of semantic similarity

tools for ontology mapping, matching, integration:

- AgreementMaker Light: `https://github.com/AgreementMakerLight/AML-Jar`
  - structural (semantic) and lexical matches
  - can use domain-specific background knowledge
- LogMap: `https://github.com/ernestojimenezruiz/logmap-matcher`
  - structural (semantic) and lexical matches
  - biology-themed versions
- NCBO Annotator: `https://bioportal.bioontology.org/annotator`
  - lexical matches only
  - can annotate full text
- recent tools and comprehensive ongoing evaluation:
  - OAEI: `http://oaei.ontologymatching.org/`

Applications of semantic similarity

semantic similarity and text mining:

- find all occurrences of classes of one (or more) ontologies in text
  - using lexical matching or semantic annotations of text
  - TextPresso (`http://www.textpresso.org/`), NCBO Annotator (`https://bioportal.bioontology.org/annotator`), WhatIzIt (`http://www.ebi.ac.uk/webservices/whatizit/info.jsf`)
  - ontology-specific text normalization tools
    - DNorm (diseases), GNorm (gene names), OSCAR (chemicals), ...
- use for database construction (automatic annotation), relation extraction, network construction (co-occurrence network), etc.

Applications of semantic similarity

- semantic similarity can be used as features in machine learning models
  - when annotation space is too large
    - e.g., GO: 50,000 classes
    - replace binary representation
  - to incorporate background knowledge
    - semantic similarity encodes *implicitly* for ontology structure and axioms
    - encodes for *specificity* of classes
  - negative: reduce all annotations to single value
    - leads to loss of information
    - but is easier to use by many machine learning methods

Summary

- many semantic similarity measures
  - graph-based
  - feature-based
- useful for similarity-based prediction
  - similar entities $\Rightarrow$ guilt-by-association
  - different entities
- combine with data and text mining
- features in machine learning methods

Acknowledgements

- Sarah Alghamdi
- Mona Alsharani
- Imene Boudellioua
- Senay Kafkas
- Maxat Kulmanov
- Fatima Zohra Smaili

Hands-on: semantic similarity

- if you have not done so *before* the tutorial, don't start now
  - you need to download *a lot* of data
  - you can just follow our demonstration and try later
  - (unless Internet is exceptionally fast for a conference Wifi, then just go ahead and do everything now)
- Jupyter Notebook
  - notebooks consist of code and rich text fragments
  - human readable (with nice figures) *and* executable
  - need to install the SciJava kernel (default: iPython)
  - very widely used
- `https://github.com/bio-ontology-research-group/ontology-tutorial`

Hands-on: semantic similarity

In the tutorial, we will

- download an ontology
- explore the ontology with OWLAPI
- classify the ontology with an OWL reasoner
  - and query using an OWL reasoner
- store the inferred version locally
- use the Semantic Measures Library to:
  - explore the ontology as graph
  - compute similarity between classes
  - use different similarity measures
  - compare patients to mice
- learn to use Onto2Vec and OPA2Vec
- you can build on this and extend for your own research!

Hands-on: semantic similarity

Do the tutorial...

Hands-on: semantic similarity

- now play with the Notebook:
  - look at the results list (check MGI)
  - try another disease (check OMIM)
  - or a drug effect (check SIDER)
- you can also test another ontology
  - GO for functional similarity
  - ChEBI for chemical (structural) similarity
  - or yeast phenotypes