OWLERY:
AN EASILY DEPLOYABLE WEB SERVICE FOR MAKING
REASONING QUERIES OVER OWL ONTOLOGIES WEB-NATIVE

HILMAR LAPP (DUKE UNIVERSITY, DURHAM, NC)
JIM BALHOFF (RENCI, CHAPEL HILL, NC)

BOSC 2018
PORTLAND, OR
DID MOST OF THE WORK:

JIM BALHOFF

https://github.com/balhoff
ONTOLOGIES ALLOW INTEGRATING ACROSS DESCRIPTIVE BIOLOGY

- Phenotypes
- Traits
- Function
- Behavior
- Habitat
- Life Cycle
- Reproduction
- Conservation Threats
Reasoning over ontologies enables linking across domains
REASONING ENABLES QUANTIFYING SHARED SEMANTICS

- morphology
  - size
  - shape
- phenotype of part of anatomical structure
- position
- quality
- phenotype of part of anatomical structure

- size phenotype of ceratobranchial bone
- size phenotype of anatomical structure
- shape phenotype of part of hyomandibular cartilage
- shape phenotype of hyomandibular cartilage
- size phenotype of ceratobranchial 5 bone
- phenotypic term

- generated phenotype
- PATO term
- subclass of phenotype

- phenotype annotation
- normalized information content

- hyomandibular condyle for the opercle; position

- Loricariidae
- ceratobranchial bone; increased width
- hyomandibular cartilage; shape
- ceratobranchial 5 bone; decreased width

- brpf1

- 24 more
- 787 more
A REASONER CAN INFER DATA IMPLIED BY OBSERVED DATA

Presence / absence of digits across Sarcopterygii

Asserted & inferred:
- 645 inferred present
- 74 asserted present
- 25 asserted absent
- 82% of taxa
High barrier to using live ontology reasoning in web-applications

OWL reasoners …

* Are written in C/C++ or Java for performance reasons
* Must classify the ontology (takes time, CPU, memory) before answering a DL query
* Are not designed for inter-process communication and client-server operation
* Don’t speak JSON or JSON-LD
Most ontology-aware applications depend on pre-reasoned data, or do not use reasoning.
Cloud-based API Paradigm for ML and AI

<table>
<thead>
<tr>
<th>Entities</th>
<th>Sentiment</th>
<th>Syntax</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>Score 0</td>
<td></td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td>Sundar Pichai</td>
<td>Score 0</td>
<td></td>
<td>PERSON</td>
</tr>
<tr>
<td>Mountain View</td>
<td>Score 0</td>
<td></td>
<td>LOCATION</td>
</tr>
<tr>
<td>Android</td>
<td>Score 0</td>
<td></td>
<td>CONSUMER GOOD</td>
</tr>
<tr>
<td>Consumer Electronic Show</td>
<td>Score 0</td>
<td></td>
<td>EVENT</td>
</tr>
<tr>
<td>users</td>
<td>Score 0.4</td>
<td></td>
<td>CONSUMER GOOD</td>
</tr>
<tr>
<td>phones</td>
<td>Score 0.7</td>
<td></td>
<td>CONSUMER GOOD</td>
</tr>
</tbody>
</table>

1. Google, headquartered in Mountain View, unveiled the new Android phone at the Consumer Electronic Show. Sundar Pichai said in his keynote that users love their new Android phones.
Owlery in a nutshell

- OWL ontology reasoner as a web-service. API similar to the OWL API de-facto reference.
- Semantic web-native: responses are in JSON-LD format, interconvertible to RDF
- Easily configurable and deployable
- Written in Scala, runs under JVM v7+
- Open source (MIT license): https://github.com/phenoscape/owlery/
Supports standard OWL serializations and reasoners

- Loads OWL ontologies in any format supported by the OWL API (RDF/XML, Functional, Manchester, Turtle)

- Can use any OWL API-compatible reasoner. Includes HermiT (OWL DL), ELK (OWL EL), and the OWL API-builtin structural reasoner
REASONER API
SIMILAR TO OWL API

RESPONSES ARE IN JSON-LD FORMAT
INCLUDES API FOR OWLET

Expands OWL class expressions embedded in SPARQL queries to FILTER expressions

https://github.com/phenoscape/owlet
**Owlery build & deployment**

- Uses the Scala Build Tool (**sbt**)

- Debian / Ubuntu server: build, then install Debian package to create a SystemV service
  
  $ sbt debian:packageBin
  
  $ sudo dpkg -i target/*.deb
  
  # to restart owlery:
  
  $ sudo service owlery restart

- Otherwise, build for native system:
  
  $ sbt stage
  
  $ cd target/universal/stage
  
  $ JAVA_OPTS="-Xmx8G" ./bin/owlery -Dconfig.file/owlery.conf
Owlery deployment using Docker

- Pre-built Docker image on Docker Hub at phenoscape/owlery
  
  ```bash
  $ docker pull phenoscape/owlery:latest
  $ docker run -v $(pwd)/conf:/srv/conf -p 8080:8080 phenoscape/owlery
  ```

- Available tags include latest (latest release), release tags (e.g., v0.13), and edge (master branch)
akka {
  loglevel = INFO
}

akka.http.server {
  request-timeout = infinite
  idle-timeout = infinite
}

owlery {
  port = 8080
  host = localhost
  kbs = [
    {
      name = uberon
      location = "http://purl.obolibrary.org/obo/uberon/ext.owl"
      reasoner = structural
    },
    {
      name = phenoscape
      location = "~/Users/jim/Desktop/owlery/phenoscape"
      reasoner = elk
    },
    {
      name = tto
      location = "~/Users/jim/Desktop/owlery/tto"
      reasoner = structural
    }
  ]
}
Summary

- Owlery provides a JSON-LD web-service API to OWL reasoning over a knowledgebase
- Ontologies constituting the knowledgebase are configurable as URLs or local files
- OWL reasoner is configurable for each input ontology
- Easily deployable using a Docker container
- Scalable through load-balancing between a pod of containers
- Actively used by Phenoscape and Virtual Fly Brain projects
Acknowledgements

- Jim Balhoff (NESCent, Phenoscape)
- OWL ontology and reasoner engineering community (including the OWL API project)
- Phenoscape personnel, PIs, and curators (http://phenoscape.org/wiki/Acknowledgments)
- National Evolutionary Synthesis Center (NESCent)
- National Science Foundation (DBI-1661456, 1661529, 1661516, 1661356; DBI-1062404 and 1062542)