Knowledge Representation & Ontology

MSc Bioinformatics

Slides: http://lokero.xyz/msc.pdf

Luke Slater (lxs511@bham.ac.uk)

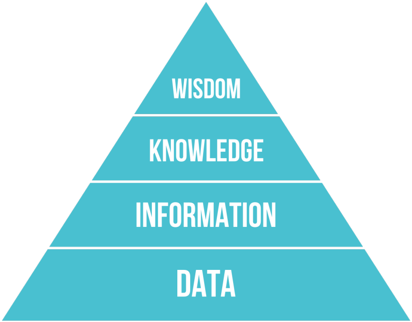
Centre for Computational Biology
University of Birmingham

28/11/2018

Objectives

- Just Semantics
- How do we express and represent knowledge?
- How can computers express and represent knowledge?
  - HTML: a degenerate form of XML
  - XML: a degenerate form of RDF
  - The Resource Description Framework (RDF)
- Ontologies
  - What are they?
  - What do they know? Do they know things? Let's find out!
  - Differences between ontologies & databases
- Afternoon: Practical

Moving from Information to Knowledge and Wisdom



- We've previously spoken about how to take some data and represent it as information and then to use it as knowledge.
- Now, we will consider how we can move, computationally, from information to knowledge and finally to wisdom.

Digression on Epistemology

- What does it mean to know something? What is knowledge? Even for humans, it is an unsolved problem.
- For a long time, Justified True Belief was regarded as an adequate solution for the question of what it means to know something. You can be said to know something if your assertion meets the following conditions:
    1. You believe the proposition is true.
    2. Your belief is somehow justified (you have a reason to believe it).
    3. The proposition itself is true.
- In 1963, Edmund Gettier released a mere 3-page paper defeating this notion by introducing the Gettier Problem...
- Nevertheless, we persist with this general idea of knowledge, for the most part.

Definition on Wisdom

Likely as many definitions as there are people who know the word, but in this case we mean: the ability to think and act using knowledge.

Symbols: Treachery of Images

Symbols and their Meanings

- This treachery lies not only in what might be called a picture, but also in words. For words are just another kind of image, or symbol.
- And so, what does it actually mean to be that which we refer when we say <u>pipe</u>?
- And how does the symbol <u>pipe</u> come to carry this meaning?
- Do these language symbols somehow directly correspond to reality (as in a picture), or are they very human mental constructs pointing vaguely at reality? (former: see Wittgenstein, latter: see later Wittgenstein).

Consensus Meaning

- These symbols are imbued with meaning by a complex cultural process of consensus, which is in a state of constant flux and ambiguity.
- We further describe both particular objects (you) and classes of objects (Human) with complicated syntactic forms, from which meaning is also derived in a cognitive process fraught with constant flux and ambiguity.
- Things are complicated further when we consider what might be said about an abstract noun, such as peace, green, or disco.

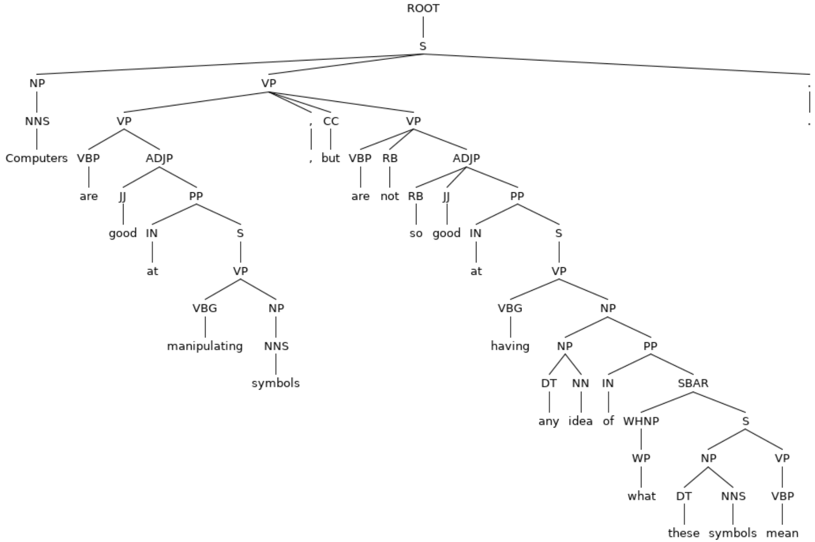
The Syntactic Web

- We all know the syntactic web: the sum of human knowledge presented in natural language, and intended for human interaction.
- As such, meaning is derived and agreed by offline consensus.
- Examples: Wikipedia, BBC news, Facebook, most websites as you know them.

Computers:

- Can count higher than you
- Needs to be a supercomputer to identify a dog

Natural Language Processing



Limitations of Classical Natural Language Processing

- Consider the sentence "I made her fish," which has at least five potential meanings (double for plurality!):
  1. I cooked a fish for her.
  2. I cooked a fish belonging to her.
  3. I created the fish she owns.
  4. I forced her to go fishing.
  5. I made her into the Platonic ideal Fish.
- All problems in NLP may be reduced to the question of ambiguity - and it is <u>everywhere</u>; in human language, symbolic equivalence is not equivalent to semantic equivalence.

Example inspired by "Speech and Natural Language Processing" 2nd ed by Daniel Jurafsky & James H. Martin.

The Limitation: Ambiguity and Meaning

- Computers are good at manipulating symbols, but are not so good at having any idea of what these symbols mean - their semantics.
- Furthermore, they struggle to translate symbols from ambiguous (read: human) languages into definite symbols to manipulate.

The Problem

- How do we make a computer understand the objects that we're describing, and the relationships between them?
- How can we make a computer know something?
- How can we gain wisdom from the knowledge we give a computer?
- Requirements:
  1. To be able to describe and to predicate upon particular entities. For example: this gibbon is very angry, or this tomato is mouldy.
  2. To be able to define the kinds of entities which may exist, and the relationships they may have between them. For example: a hand may have fingers, or a parent must have a child.

Separating the entity and the instance

- This separation of the class and the particular is along the lines of the Plato's Theory of Forms, or more generally Idealism: that particular objects (a chair) are partaking in a pure form which idealises and represents the most accurate reality of that object (The Chair or Chairness).
- This is one solution to the metaphysical problem of Universals.
- While rife with philosophical problems (much like any alternative theory), it happens to be extremely useful to us: computer scientists may notice a strong correlate when considering Object Orientated Programming: Classes and Instances.

The Semantic Web

- Utilises the infrastructure of the world wide web as a data communication methodology.
- Instead of only presenting data: link, interpret, and analyse data.
- In the consumer sphere this forms the basis for the 'Internet of Things.'

Describing instances

Almost accidentally, the means by which we could develop a semantic framework for knowledge representation was already in heavy use throughout the world wide web.

HTML

Web pages are created in HyperText Markup Language (HTML)

```
<div id="luke">
  <h2>Bands Luke likes:</h2>
  <ul id="bands">
    <li>Pavement</li>
    <li>Modest Mouse</li>
    <li>Built to Spill</li>
    <li>Japanese Breakfast</li>
  </ul>
</div>
```

Bands Luke likes:

- Pavement
- Modest Mouse
- Built to Spill
- Japanese Breakfast

Natural Language Inference

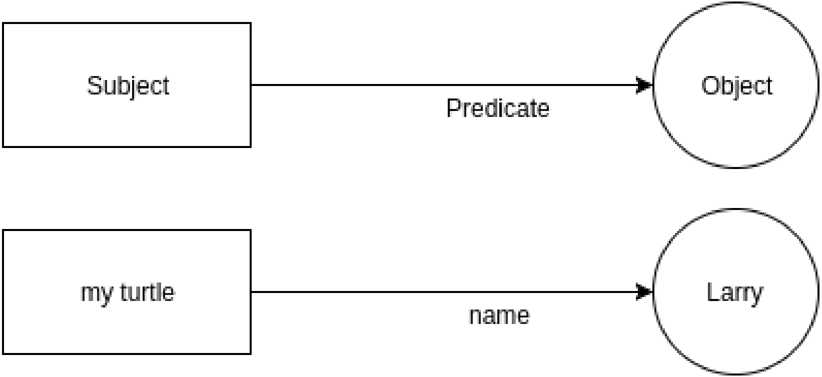
- The assertions we can infer from this text are numerous:
  1. A person exists.
  2. A person is named Luke.
  3. There is at least one person.
  4. The person Luke likes some bands.
  5. At least four bands exist.
  6. Some people like bands.
  7. Pavement, Modest Mouse, Built to Spill, Japanese Breakfast are bands.
  8. Luke stands in a relationship of likes with all these bands.
  9. Some bands are liked.
  10. At least one person likes Pavement, Modest Mouse, Built to Spill, and Japanese Breakfast.

HTML: A Degenerate Form of XML

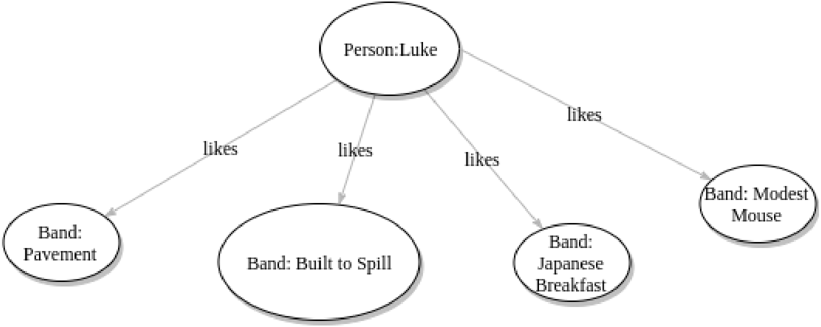
- HTML was later generalised, to create the eXtensible Markup Language (XML), without a focus on describing how a document should be rendered visually.
- This is a more general markup language, which can organise and "mark up" or extend any kind of textual data with any additional information.

RDF: An Exhilirate Form of XML

- A method of representing structured data for the semantic web.
- Used heavily within science for experimental results, datasets and various open data, particularly in the biological and biomedical domains.



RDF: Semantic Networks



Wikipedia vs DBPedia

Aberystwyth

From Wikipedia, the free encyclopedia

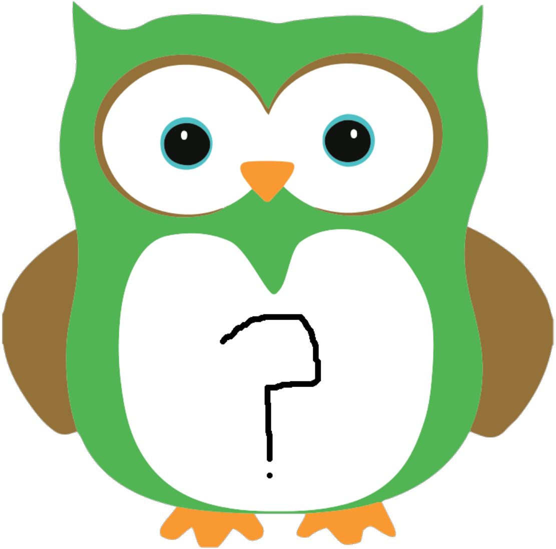
Aberystwyth (Mouth of the Ystwyth, /ˌæbəˈrɪstwɪθ/, Welsh: [abɛrˈəstʊɨθ]) is a historic market town, administrative centre, and holiday resort within Ceredigion, West Wales, often colloquially known as **Aber**. It is located near the confluence of the rivers Ystwyth and Rheidol.

Historically part of Cardiganshire, since the late 19th century, Aberystwyth has also been a major Welsh educational centre, with the establishment of a university college there in 1872. At the 2001 census, the town's population was 15,935;[3] it was reduced to 13,040 at the 2011 Census. During nine months of the year, there is an influx of students—to a total number of 10,400 as of September 2012.[4] Including the suburbs of Llanbadarn Fawr, the population is 16,420.

Contents [hide]

dbo:areaCode	• 01970
dbo:ceremonialCounty	• dbr:Dyfed
dbo:country	• dbr:United_Kingdom • dbr:Wales
dbo:distanceToCardiff	• 144840.960000 (xsd:double)
dbo:gridReference	• SN585815
dbo:populationTotal	• 13040 (xsd:integer)
dbo:postalCode	• SY23
dbo:principalArea	• dbr:Ceredigion
dbo:thumbnail	• wiki-commons:Special:FilePath/Aberystwyth_shore.jpg?width=300

What is an ontology, anyway?



Ontology: Describing the objects

- Ontologies categorise, define and relate the kinds of things in a domain.
- Easily thought of as the 'schema' for the data represented by RDF (and in any other format).
- Exact definitions differ, but most ontologies share four main features:
  1. Classes and relations.
  2. Domain vocabulary.
  3. Metadata and descriptions.
  4. Axioms and formal definitions.

Ontology: Classes (and relations)

- A class is a category of things which can exist within a particular world, defined by the necessary and sufficient conditions for a thing to exist within that category (intensional definition).
- Each distinct class has its own IRI (Internationalised Resource Identifier), but may share labels, descriptions, or any other information with other classes.
- This does not necessarily make them semantically equivalent!
- e.g. HP:0002457 and MP:0000436 both have the label 'abnormal head movements'

Ontology: (Classes and) relations

- Also concerned with the relationships between things. A fundamental relationship between classes is the is-a relationship.
- All ontologies start with the root class Thing, and all other classes in the ontology stand in a transitive subclass relationship to it.
- In this way, they form, at a most basic level, a taxonomy of categorisation for a particular universe of interest.
- Other common relationships are: part-of, inheres-in, caused-by, regulates.
- Many relationships carry deeply argued issues and side-effects: fingers!

Ontology: Domain Vocabulary

- Structured vocabulary for a domain
- Usually provided in the form of short 'labels' for each class (can have multiple)
- Provision for translated terms for classes
- Synonyms; the same category of things in a different context
- Together with the structure of the ontology, we gain a large and well-organised set of relevant terms for our domain
- Reduces ambiguity and misunderstanding in the conceptualisation for a domain

Ontology: Domain Vocabulary

- '[X]External causes of morbidity and mortality'
  - '[X]Complications of medical and surgical care'
  - '[X]Event of undetermined intent'
  - '[X]Injury - self-inflicted'
  - '[X]Legal intervention and operations of war'
  - [X]Mugged
  - '[X]Other external causes of accidental injury'
    - '"[X]Exposure to smoke'
    - "[X]Overexertion
      - '[X]Lack of water'
      - '[X]Overexertion and strenuous or repetitive movements'
      - '[X]Prolonged stay in weightless environment'
        - **'[X]Prolonged stay in weightless environment occurrn on farm'**

Ontology: Metadata and Descriptions

- Descriptions, usually provided using the genus-differentia method.

  temporal pattern
  : The speed at which disease manifestations appear and develop.

  insidious onset
  : Gradual, very slow onset of disease manifestations

- Database cross-links, providing an assurance of semantic equivalence to data about an entity from an external source.

Ontology: Axioms and Formal Definitions

- The formal aspect of ontologies, axiomatic expression of meaning.
- Example: hypertension defined as has part some ((blood and (part-of some arterial system)) and (has-quality some ((increased pressure and (has-modifier some chronic)) and (has-modifier some abnormal))))
- Enable semantic analysis and (ostensibly) cross-species integration of knowledge.

Ontology Reasoning

- Reasoning is the use of a classifier to evaluate all logical consequences of the explicit statements in an ontology.
- This allows us to infer new knowledge, and evaluate the consistency of our logical model (and can tell us why in the case of inconsistency).
- There are many different reasoners, which use slightly different methods and support different subsets of description logic (they are often said to have different levels of 'expressivity').

How do ontologies differ from databases?

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

Web Ontology Language (OWL)

- Poor acronym, reference to Winnie the Pooh.
- Based on description logics, a fragment of first order logic.
- A set of language specifications based on description logics.
  - **Full** Every construct in OWL is available. Fully undecidable to reason. Cardinality restrictions, domain, range, and value restraints.
  - **EL** Subset of OWL which is guaranteed to be classifiable in polynomial time. Only way to reach inconsistency is through disjointness.
  - **RL** A subset which mirrors the features of relational databases.
- Less expressive reasoners can simply ignore the more expressive content within ontologies.

- `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`: $X \overset{\equiv}{\leftrightarrow} Y$, $\{X, Y\}$

OWL Syntax

- originally an extension of RDF and RDF Schema
- Several different (mostly) 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
$\{a\} \sqcup \{b\} \sqcup \ldots$	{a b ...}	{John Robert Mary}

Basic Inference Example

1. All meat comes from animals. (meat comes-from animal)
2. All beef is meat. (beef is-a meat)
3. Therefore, all beef comes from animals.

A simple subsumptive relationship, if the first two statements are true (note: our ontology isn't concerned with the truth of the statements), then the conclusion is necessarily true. So, our reasoner will add this logical statement to the ontology (beef comes-from animal)

Basic Inconsistency Example

- Reptile is-a NOT Mammal
- Person is-a Mammal
- Turtle is-a Reptile
- Turtle is-a Person

Because a turtle is necessarily a reptile, and a reptile is necessarily not a mammal, a turtle cannot be both a reptile and a person.

Questions