Lecture Week 1

DIKW pyramid

1. **Data -** raw existence, any form, may or may not be usable
2. **Information -** data given meaning by relational connections, is contained in descriptions, answers simple questions
3. **Knowledge** - collection of related information, with intent of being useful
   It is information enriched with semantics (and possibly past experience)
4. **Wisdom** - ability to make sound judgements and decisions (evaluated understanding)

What is the relationship between data, information and knowledge?

Data becomes information by convention (adding standard syntax), information becomes knowledge by cognition, and knowledge becomes wisdom by contemplation.

Formal Knowledge Representation: Ontologies

Keys for Speaking a Common Language:

1. Syntax
2. Semantics
3. Taxonomy - classification of concepts
4. Thesauri - relations b/w concepts
5. Ontologies - rules and knowledge about which relations are allowed and make sense

Properties of a Language:

1. Symbols (spoken/written/manual)
2. Convey meaning
3. Means of expression
4. Means of communication

Correct Interpretation depends on:

<u>For Successful Communication:</u>

1. Syntax (arrangement of symbols, set of rules, normative structure of data)
2. Semantics (meaning of the symbols of language within a context and rules of syntax)

<u>For Understanding</u>

3. Context (relationship with surrounding expressions and sender/receiver of the message)
4. Pragmatics (intended purpose of communicator, varies in different situations/context)

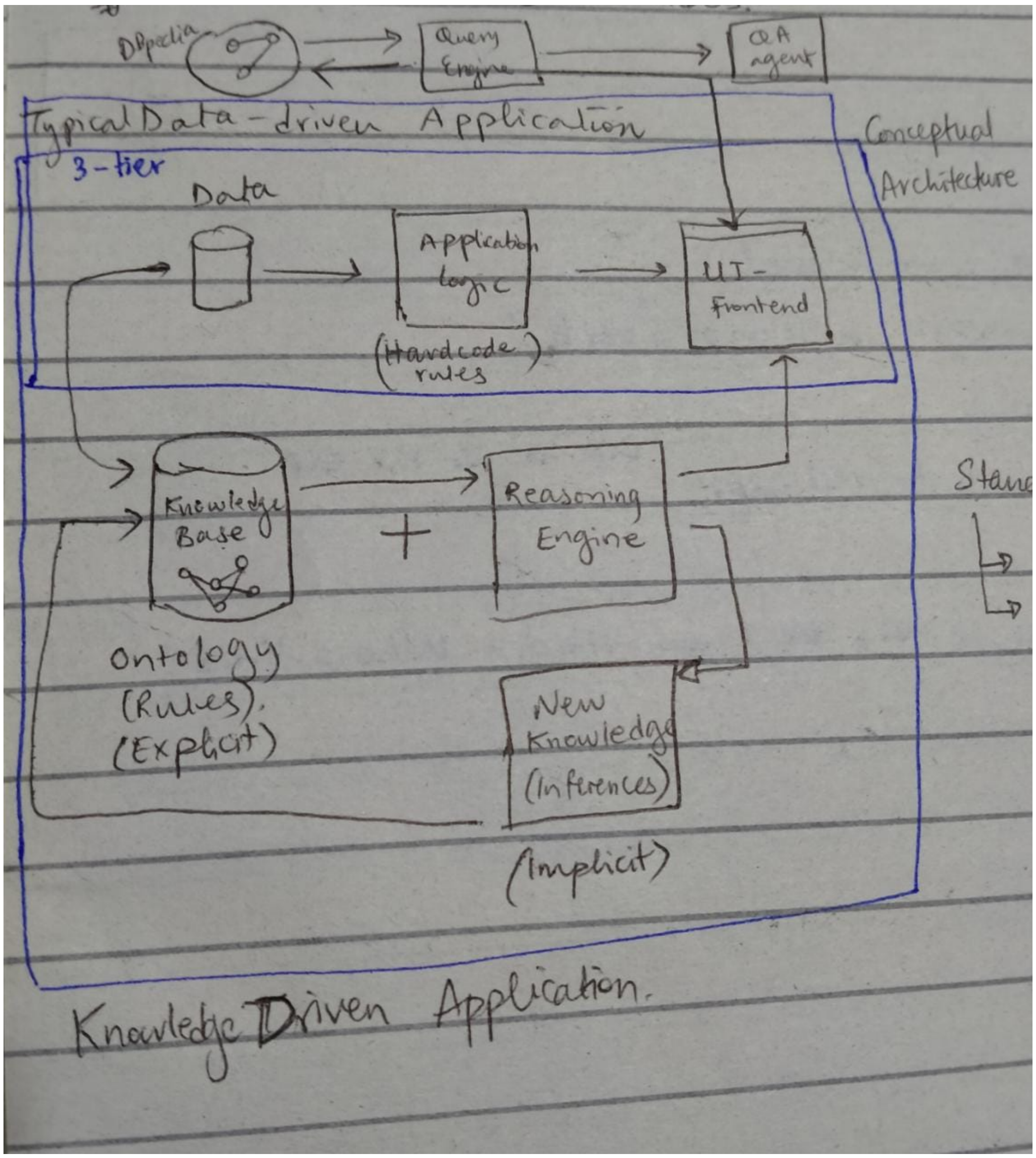
<u>Context Depends On</u>

5. Experience (common sense/world knowledge)

Lecture Week 2-3

* The meaning (Semantics) of information is expressed with the help of knowledge representations (Ontologies).

Difference b/w typical Data-Driven and Knowledge-Driven Applications



Difference between the Internet, the Web, and Search Engines:

-The Internet created a standard way for computers to exchange information with one another.

- The Web is a huge document and information storage and retrieval system.

-> The Web also created a syntax called **HTML** for displaying information.

-> When you enter a website address, the website retrieves the document present at the address and sends it back to your web browser.
- A Search Engine is a software web-based tool that enables users to locate information on the World Wide Web.

Web 1.0 vs 2.0 vs. 3.0:

Web 1.0: The Internet
- Computer centered processing
- Web of documents
- Static websites
- Goal: information sharing
- HTML

Web 2.0: The World Wide Web
- Document-driven
- Dynamic web applications
- Application-level **interoperability**
- Information retrieval via search engines
- XML

Web 3.0: The Semantic Web
- Data-driven
- Decentralized global database (knowledge base)
- Focus on relating knowledge
- Knowledge represented by Ontologies
- RDF, OWL

WWW to Web of Data/ Semantic Web: content can be **read** and **interpreted** correctly by machines. It requires use of:
- NLP technologies
- Semantic Web technologies

Syntactic vs Semantic Search:

Syntactic: key-words - doesn't consider natural language ambiguity e.g synonyms, metaphors
Semantic: knowledge-driven (by understanding meaning)

Limitations of the traditional web:
- Data Silos: no data level linkages
- Computers blindly retrieve information; they don't understand the information.
- They understand the syntax, not semantics
- The traditional web was document-based
- Searches depend on keyword matching

What is the semantic web? Give another name for semantic web.
- also called the **Web of Things/Data** (people, events, movies, documents, any concept).
- helps computers understand the **meaning** behind a web page
- establishes relations between **things** on the Internet.
- searches not dependent solely on keyword matching, rather on semantics

* Whereas you represent a **document in HTML**, you represent **data in RDF**.

How is meaning expressed on the semantic web?

RDF
- expresses meaning with encoding in triplets of {subject, predicate, object}
- Subjects + predicates are necessarily identified by URIs, while objects may/may not be

Give three inherent benefits of the semantic web.
1. Meaning of data is processed and understood
2. Semantics-based interoperability - possible to **integrate** and **interrelate** heterogenous data
3. Auto generation of knowledge through Reasoning engine and Knowledge Base (Ontology/Rules)

What are the key elements in the first three layers of the semantic web stack?
1. **URI** for identification & **URL** for address
2. **HTTP** for Communication protocol
3. **RDF**

What is the linked data cloud?

Where a variety of data sources are published in **unified format (RDF)** and **linked** on Cloud. This removes the problem of **Data Silos**.

Linked Data Principles:
1. Use URIs to name things/concepts
2. Use HTTP URIs (to make them locatable)
3. Provide useful information against each URI when looked up in standard formats (RDF, SPARQL)
4. Include links to other URIs for more discovery

Three rules for putting something on the web:
1. All data/concepts have names starting with **http** (for searching)
2. Data is returned with a standard format
3. Data has relationships with other data.

Tim Berners-Lee's 5-Star Criteria for Linked Open Data:

1. Data available on the Web with open license
2. 1 + Data in machine-readable structured data (e.g Excel, not image scan)
3. 2 + non-proprietary format (e.g. CSV, not Excel)
4. 3 + RDF and SPARQL for identifying things on the Web
5. 4 + link your data to other data sources (creates context)

Lecture Week 3-4

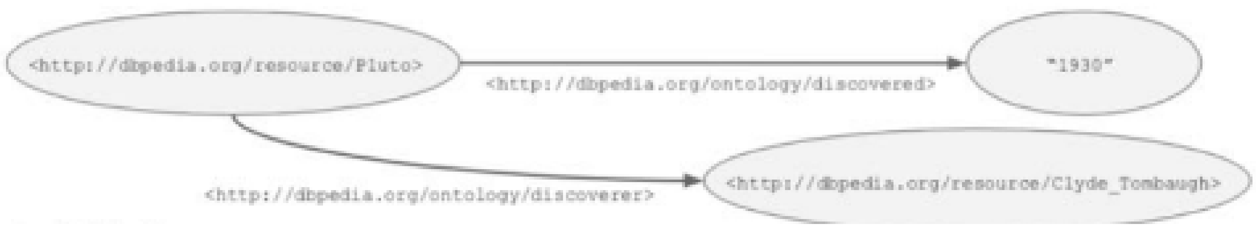
What is considered the nucleus of a linked data cloud?
DBpedia

What is DBpedia?

N-Triples Notation

- URIs/IRIs in angle brackets
- Literals in quotation marks
- Triple ends with a period

Difference between N-Triples, RDF/XML and Turtle Notations



<u>N-Triples</u>

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930

<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discoverer>
<http://dbpedia.org/resource/Clyde_Tombaugh> .
```

<u>RDF / XML</u>

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ns0="http://dbpedia.org/ontology/">
  <rdf:Description rdf:about="http://dbpedia.org/resource/Pluto">
    <ns0:discovered>1930</ns0:discovered>
    <ns0:discoverer rdf:resource="http://dbpedia.org/resource/Clyde_Tombaugh"/>
  </rdf:Description>
</rdf:RDF>
```

Note:

- rdf:about=”........” (subject)
- If object is a literal, it is padded with an opening and ending predicate tag
  <ns0:discovered>1930</ns0:discovered>
- If object is a resource (URI), the URI lies within the predicate’s tag
- <ns0:discoverer rdf:resource=”http://dbpedia.org/resource/lyde_Tombaugh” />

<u>Turtle</u>

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@base  <http://dbpedia.org/resource/> .

<Pluto> dbo:discovered "1930" .

<Pluto> dbo:discoverer <Clyde_Tombaugh> .
```

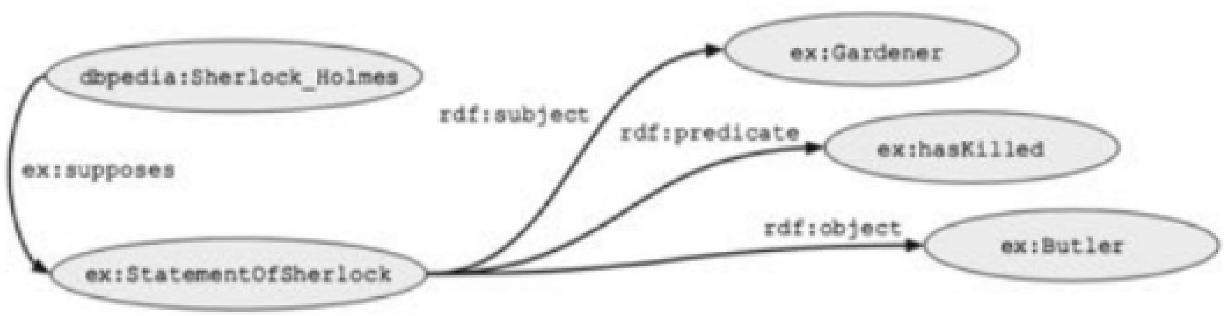
Lecture Week 4-5

RDF - means of expressing Knowledge

- **Resource**: can be anything uniquely identifiable and referenceable via URI
  It's an **IRI** (International Resource Identifier)
- **Description**: representing properties+relations via graphs
- **Framework**: web based protocols (URI,HTTP,XML,Turtle,...) based on formal model (semantics)

RDF Reification: making statements about statements
Example

- Sherlock Holmes supposes that the gardener has killed the butler



@prefix	rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
@prefix	dbpedia <http://dbpedia.org/resource/> .
@prefix	ex: <http://example.org/Crimestories#> .

Turtle Notation
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/Crimestories#> .

dbpedia:SherlockHolmes ex:supposes ex:StatementOfSherlock .
ex:StatementOfSherlock a rdf:Statement ;
rdf:subject ex:Gardener ;
rdf:predicate ex:hasKilled ;
rdf:object ex:Butler .

- RDFS enhances the expressivity of RDF.
- RDF Schema: officially the "RDF Vocabulary Description Language"
- RDFS enables the following:

    - definition of classes via rdf:Class
    - class instantiation in RDF via rdf:type
    - Definition of properties via rdf:Property + property restrictions (rdfs:domain, rdfs:range)

- Everything in the RDF model / schema is a resource.
- Definition of hierarchical relationships:
    - Subclasses, superclasses (rdfs:subClassOf)
    - Subproperties, super properties (rdfs:subPropertyOf)

RDFS Semantics:

- Semantics of a term from an RDFS ontology is given in terms of its properties and its instances
- RDFS is a data definition language based on formal semantics
- Formal semantics enables RDF(S) to draw valid and sound logical inferences

Conclusions we can deduce using RDF(S)

- Deduction of entity class membership from domain of one of its properties
- Deduction of entity class membership from the range of one of its properties
- Deduction of entity superclass membership from a class hierarchy.
- Deduction of entity new facts from subproperty relationships.