Semantic Web, Knowledge Graphs, and Semantic Web Technologies

Exam-Ready Notes

Contents

1 Big Picture: Why We Need the Semantic Web

1.1 The Problem with Today's Web

Current web = web of documents. Web pages (HTML) are written mainly for humans to read, not for machines to understand. [file:1][file:3]

- A page about "Bob" may contain his name, job, birthdate, university, etc.
- You, as a human, can understand: Bob is a person, works at Heriot-Watt, is interested in AI.
- A computer only sees text and HTML tags like `<p>`, `<b>`, etc. [file:3]

Key limitations: [file:3]

- Documents are not *machine-understandable* (only machine-displayable).
- No standard way for machines to know what *Bob*, *works at*, *Heriot-Watt* actually *mean*.
- Hard to automatically combine data from multiple websites or services.

1.2 Data, Information, Knowledge (DIKW)

From the introduction slides: [file:1]

- **Data**: raw facts with no (or little) context.
  - Examples: numbers, temperatures, prices, text strings.
- **Information**: data placed in context.
  - Example: daily sales figures organised by date and shop.
- **Knowledge**: rules, patterns, and understanding that we can use to make decisions.
  - Example: "If sales drop three days in a row, launch a promotion". [file:1]

Traditional computer systems focus on *data* and *information*, but the semantic web + knowledge graphs aim to bring explicit *knowledge* closer to machines. [file:1][file:3]

1.3 From Web of Documents to Web of Data

Idea: turn the web into a large machine-readable graph of data (web of data). [file:3]

- Instead of just linking HTML pages, we link *entities* (people, places, events, things) and their relationships.
- This requires a standard way to describe entities and relationships so machines can interpret them.

1.4 Real-World Motivating Examples

Search engines (e.g., Google).

- When you search "Eiffel Tower", you get a knowledge panel with:
  - Location, height, construction dates, architect, etc.
- This is powered by a *knowledge graph* (Google Knowledge Graph). [file:3]

Recommendation systems (e.g., Amazon, Netflix).

- Connects users, items, genres, ratings, etc., in a graph.
- Allows reasoning like: "Users who liked this also liked ..."

Government and Open Data.

- Data.gov.uk example: combining crime data, environment, services to produce personalised local newspapers. [file:3]
- Makes public data understandable and useful for citizens and applications.

2 Semantic Web and Web of Data

2.1 What is the Semantic Web?

Definition (simplified from slides): [file:3]

- **Semantic Web** = a "web of data" that lets machines understand the *meaning* (semantics) of information on the web.
- Achieved by inserting machine-readable metadata about pages and describing how data items are related.

  Goal: [file:3]

- Allow automated agents (programs) to:
  - Retrieve data from many sources.
  - Understand how pieces of data relate.
  - Perform tasks on behalf of users.

2.2 Semantic Web vs Syntactic Web

- **Syntactic web**: about documents (HTML), mainly for human reading. [file:3]
- **Semantic web**: about *things* (people, organisations, places, events) and *relationships* between them. [file:3]

2.3 Example: Adding Context with Metadata

Slides show an example for the person "Radu Mihailescu". [file:3]

- Human reader infers:
  - It is a human.
  - Name: Radu Mihailescu.
  - Works at Heriot-Watt University.
  - Research domains: Machine Learning, Deep Learning, AI, Optimisation, etc.
- A machine cannot infer this spontaneously.

  By adding metadata:

```
<human/>
<male/>
<name>Radu Mihailescu</name>
<academic/>
<workAt>
  <organisation>
    <type>university</type>
    <name>Heriot-Watt</name>
  </organisation>
</workAt>
<researchProfile>
  <link>https://scholar.google.com/.../RaduMihailescu</link>
</researchProfile>
```

machines now have explicit structure they can process. [file:3]

2.4 Semantic Web and Linked Data Principles

Linked Data principles (Tim Berners-Lee). [file:3]

- Use IRIs to uniquely identify things (entities).
- Use HTTP IRIs, so they can be looked up on the web.
- Provide useful information about the thing:
  - HTML for humans.
  - RDF for machines.
- Include links to other related things, so data can be discovered.

Semantic Web ⇒ Web of Data.

- A web that implements the semantic web principles is called the *web of data*. [file:3]

3 Graphs and Knowledge Graphs

3.1 What is a Graph?

Definition from slides: [file:3]

- A graph is a set of:
  - **Vertices** (or *nodes*).
  - **Edges** (connections between nodes).
- Can be:
  - **Undirected**: edges have no direction.
  - **Directed**: edges have a direction (arrow).
- Nodes and edges can have labels and properties (e.g., weights).

3.2 Graph Examples (Intuitive)

- **Map graph**: cities = nodes, roads = edges. [file:3]
- **Social network graph**: people = nodes, "friend of" = edges. [file:3]
- **Transport network**: stations = nodes, routes = edges.

3.3 What is a Knowledge Graph?

Knowledge base (from multiple definitions). [file:3]

- Store of information or data that a system can use to solve problems.
- Expressed using a formal knowledge representation language.

Knowledge graph.

- A **knowledge graph** is a knowledge base that uses a graph data model. [file:3]
- Typically directed, labelled graphs, where nodes represent entities and edges represent relationships.

3.4 Knowledge Graph Example (Bob and Alice)

Slide example: [file:3]

- Alice is a friend of Bob.
- Bob is a Person.
- Bob was born on 14 July 1990.
- Bob is interested in the Mona Lisa.
- The Mona Lisa was created by Leonardo da Vinci.
- "La Joconde à Washington" is about the Mona Lisa.

Textual diagram (you can later convert this to TikZ if you like):

```
Alice  --friendOf-->  Bob  --type-->  Person
                          --birthDate--> 14-07-1990
                          --interestedIn--> MonaLisa
MonaLisa --createdBy--> LeonardoDaVinci
MonaLisa --subjectOf--> "La Joconde à Washington"
```

3.5 Customer Knowledge Graph Example

Another slide example: [file:3]

- **Customer A**
  - First name: Timothy.
  - Last name: Nguyen.
  - Born: 1953.
  - Lives at: 546 Drury Lane.

- Initiated transaction #34R56Q.

- **Address:** 546 Drury Lane is in Boston, MA.
- **Transaction #34R56Q**
  - Purchased Product #312.
  - Occurred on March 9, 2020.
  - Occurred at Store #31.
- **Date & Store:** Both are in Boston, MA. [file:3]

This can be drawn as a directed graph: Customer → Transaction → Product; Customer → Address → City; etc.

3.6 Who Uses Knowledge Graphs?

From ISWC citation in slides: [file:3]

- Major companies: Google, Yahoo!, Microsoft, Facebook.
- Use cases:
  - Semantic search (Google Knowledge Graph, Microsoft Satori).
  - AI assistants (e.g., Amazon Information Graph).
  - Competitive analytics, reputation analytics (e.g., Palantir, Quid).

Example public knowledge graphs: [file:3]

- Wikidata.
- DBpedia.
- Freebase (historical).
- YAGO.
- WordNet, ConceptNet.

3.7 Knowledge Graph Construction (High-Level)

Two main tasks: [file:3]

1. **Identify entities** (Named Entity Recognition).
   - Find objects/individuals in text (e.g., "John", "Dubai", "Heriot-Watt University").
   - Disambiguate strings ("Paris" the city vs. "Paris" the person).
2. **Identify relationships**.
   - Extract links between entities: "John teaches Big Data Management", etc.

4 Open Data, Linked Data, and Linked Open Data

4.1 Definitions

From the introduction slides: [file:1][file:3]

- **Big data**:
  - High *volume*, *velocity*, *variety* of data.
  - Requires new processing techniques for insights.
- **Open data**:
  - Data available for reuse free of charge, under an open licence.
- **Linked data**:
  - Best practices for publishing structured data on the web using IRIs, RDF, and links. [file:1][file:3]
- **Linked open data (LOD)**:
  - Open data (with free licence) that is also linked using linked data principles.
- **Linked closed data**:
  - Linked data, but without a free/open licence (e.g., internal enterprise data). [file:3]

4.2 Laney's 5Vs for Big Data

Characteristics of big data (from slides): [file:1]

- **Volume**: large size of data.
- **Velocity**: speed at which data is generated/collected.
- **Variety**: different data types (structured, semi-structured, unstructured).
- **Veracity**: data quality, trust, provenance.
- **Value**: ability to derive insights and decisions.

4.3 Five-Star Linked Open Data Scheme

Tim Berners-Lee's 5-star scheme (slides): [file:3]

- **1**:
  - Data available on the web, in any format, under an open licence.
  - Example: PDF report.
- **2**:
  - Data available in a machine-readable, structured format.
  - Example: Excel (XLS) instead of scanned image. [file:3]
- **3**:
  - Use non-proprietary (open) formats.
  - Example: CSV instead of Excel. [file:3]

- **4**:
  - Use open standards like URI/IRI and RDF; SPARQL queries possible. [file:3]
- **5**:
  - Link your data to other data to provide context; full semantic interoperability. [file:3]

4.4 LOD Growth

Slides show LOD cloud growth: [file:3]

- May 2007: 500 million RDF triples, 120,000 RDF links.
- September 2011: 31.6 billion triples, 503 million links.
- By 2015: number of sources multiplied by 4 (compared to 2011).

Key idea: LOD has grown from a small set of datasets to a huge global cloud of interlinked data.

5 Semantic Web Technology Stack

5.1 Overview of the Stack

The W3C semantic web stack consists of layers, each building on the lower ones. [file:2]

- **Identifiers / Namespaces**: IRIs, CURIEs.
- **Serialisation / Formatting**: XML, JSON, etc.
- **Data model**: RDF.
- **Vocabulary schema**: RDF Schema (RDFS).
- **Ontology modelling**: OWL.
- **Logic and rules**: SWRL and other rule languages.
- **Proof & trust**: digital signatures, provenance (often not covered in detail).

5.2 IRIs and Namespaces

IRI.

- IRI = Internationalized Resource Identifier. [file:2]
- Generalisation of URI that supports non-ASCII characters (e.g., Chinese, Japanese).
- Each IRI denotes a resource (thing).

Examples: [file:2]

- `http://example.com/resource/Bob`
- `http://.`
- `http://example.com/.html`

Namespaces.

- Help avoid name collisions when different organisations define terms with the same local name. [file:2]
- A namespace is usually a common IRI prefix:
  - `http://myonto.hw.ac.uk/business#`
  - `http://myonto.hw.ac.uk/software#`
- Example:
  - `http://myonto.hw.ac.uk/business#developer`
  - `http://myonto.hw.ac.uk/software#developer`

CURIE (Compact URI).

- Shortened form of IRI: [file:2]
- Examples:
  - `isbn:0393315703`
  - `doi:10.1038/sdata.2016.18`

5.3 RDF: Resource Description Framework

5.3.1 RDF Data Model

RDF is a data model used to describe resources and their relationships. [file:2]

- Core idea: **triples**.
- A triple is of the form: (s, p, o) = (subject, predicate, object). [file:2]
- **Subject** (s):
  - The resource (entity) being described.
  - Usually an IRI.
- **Predicate** (p):
  - The property (relationship type).
  - Always an IRI.
- **Object** (o):
  - Either a resource (IRI) or a literal (text, number, date).

An RDF dataset corresponds to a directed labelled graph: [file:2][file:3]

- Nodes = subjects and objects (IRIs and literals).
- Edges = predicates (properties).

5.3.2 RDF vs HTML vs XML

Slides compare them: [file:2]

- **HTML**:
  - Markup controls how to display text.
  - Example: `<p><b>HTML</b> merely tells ...</p>`
  - Computer does not know semantics of the words.
- **XML**:
  - User defines tags:

    ```
    <name>
      <given>Ken</given>
      <family>McLeod</family>
    </name>
    ```

  - Structure may suggest meaning but XML itself does not provide global semantics.
  - No standard way to tell other systems what the tags mean. [file:2]
- **RDF**:
  - Designed explicitly to encode *meaningful relationships* between resources.
  - Globally interpretable by any RDF-aware tool.

5.3.3 RDF Example (Informal and Formal)

From the slides: [file:2][file:3]

- Informal statement: `<Bob> <Knows> <Alice>`
- With namespaces (Turtle-like):

```
PREFIX ex:   <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

ex:bob#me foaf:knows ex:alice#me .
```

RDF components compared to object model. [file:2]

- Resource ≈ Object.
- Property ≈ Attribute/Field.
- Attribute value can be:
  - Another resource (object reference).
  - Literal (int, string, date, etc.).

5.4 RDFS and OWL (Schemas and Ontologies)

5.4.1 RDF Schema (RDFS)

RDFS (RDF Schema) adds basic vocabulary for describing: [file:2]

- Classes (types of things).
- Subclass relationships.
- Domains and ranges of properties (what types they connect).

This lets you define simple *schemas* for your RDF data.

5.4.2 OWL: Web Ontology Language

OWL builds on RDF/RDFS and supports more expressive modelling. [file:2]

- Allows defining and using your own:
  - Classes (concepts).
  - Properties (relationships).
- Adds richer constructs:
  - Class hierarchies (subclasses).
  - Property characteristics (functional, inverse, transitive).
  - Cardinality constraints (e.g., exactly one birthdate).
  - Equivalence, disjointness, etc.
- The resulting model is called an **ontology**. [file:2]

Ontologies vs other vocabularies. [file:2]

- **Folksonomy**: community-chosen tags, no formal relationships.
- **Controlled vocabulary**: list of predefined terms, still no structured relationships.
- **Taxonomy**: controlled vocabulary + hierarchy (is-a tree).
- **Thesaurus**: taxonomy + additional relations like synonyms.
- **Ontology**: terms + rich relationships + ability to define constraints and rules.

5.5 Logic, Inference, and Open World Assumption

5.5.1 Logic Layer

The logic layer allows you to write rules on top of the ontology. [file:2]

- Example rule: "All humans are mortal".
- Together with "Socrates is a human" you can infer: "Socrates is mortal".
- Languages: SWRL (Semantic Web Rule Language), others. [file:2]

5.5.2 Inference Examples

Given: [file:2]

- Socrates is a human.
- All humans are mortal.

A reasoner can infer the unstated fact:

- Socrates is mortal.

5.5.3 Open World vs Closed World

Closed World Assumption (CWA).

- Used in traditional relational databases.
- Anything not explicitly stored is assumed **false**. [file:2]

Open World Assumption (OWA).

- Used in semantic web and OWL reasoners.
- Anything not known may be *true or false*, but is treated as **unknown**. [file:2]
- Example:
  - We know: Socrates is mortal.
  - We know nothing about John.
  - Question: Is John mortal?
    - CWA: false.
    - OWA: unknown (maybe we just do not have that information yet).

Why OWA is important.

- The web is inherently incomplete; we cannot assume that data we do not see does not exist.

5.6 RDF Serialisation Formats

RDF is an abstract graph model. To store it in files or send it across the network, we use concrete syntaxes. [file:2]

5.6.1 RDF/XML

- Original W3C standard serialisation. [file:2]
- XML-based, so can be processed by standard XML tools.
- Example (from slides):

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:schema="http://schema.org/">
  <rdf:Description rdf:about="http://example.org/bob#me">
```

```
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <schema:birthDate
      rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
      1990-07-04
    </schema:birthDate>
    <foaf:knows rdf:resource="http://example.org/alice#me"/>
    <foaf:topic_interest
      rdf:resource="http://www.wikidata.org/entity/Q12418"/>
  </rdf:Description>
  ...
</rdf:RDF>
```

5.6.2 N-Triples

- Simple line-based plain text. [file:2]
- Each line = one triple.
- Uses full IRIs in <>.
- Example:

```
<http://example.org/bob#me>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/bob#me>
  <http://xmlns.com/foaf/0.1/knows>
  <http://example.org/alice#me> .
```

5.6.3 N-Quads

- Extension of N-Triples with a *fourth* element: graph IRI. [file:2]
- Each line: subject, predicate, object, graph.
- Example:

```
<http://example.org/bob#me>
  <http://xmlns.com/foaf/0.1/Person>
  <http://example.org/bob> .
```

5.6.4 Turtle

- Human-friendly, compact syntax. [file:2]
- Supports namespaces with PREFIX.
- Example:

```
BASE   <http://example.org/>
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX schema: <http://schema.org/>
PREFIX dcterms:<http://purl.org/dc/terms/>
PREFIX wd:     <http://www.wikidata.org/entity/>
```

```
<bob#me> a foaf:Person ;
  foaf:knows <alice#me> ;
  schema:birthDate "1990-07-04"^^xsd:date ;
  foaf:topic_interest wd:Q12418 .

wd:Q12418 dcterms:title "Mona Lisa" ;
  dcterms:creator <http://dbpedia.org/resource/Leonardo_da_Vinci> .
```

5.6.5 TriG

- Extension of Turtle for named graphs. [file:2]
- Uses `GRAPH <graphIRI> { ... }`.

5.6.6 JSON-LD

- JSON-based format for RDF. [file:2]
- Favoured by search engines (e.g., Google) for web page markup.
- Embedded using `<script type="application/ld+json">` in HTML.

5.6.7 RDFa

- Embeds RDF metadata inside HTML attributes. [file:2]
- Core attributes: `about`, `resource`, `property`, `typeof`, `rel`, `rev`.
- Example snippet:

```
<body prefix="foaf: http://xmlns.com/foaf/0.1/
              schema: http://schema.org/
              dcterms: http://purl.org/dc/terms/">
  <div resource="http://example.org/bob#me" typeof="foaf:Person">
    <p>
      Bob knows <a property="foaf:knows"
        href="http://example.org/alice#me">Alice</a>
      and was born on the
      <time property="schema:birthDate">1990-07-04</time>.
    </p>
  </div>
</body>
```

5.7 Advanced RDF Modelling Patterns

5.7.1 Integrating Graphs from Multiple Sources

- Two RDF graphs from different datasets can be combined if they share the same IRIs. [file:2][file:3]
- Example:
  - Both graphs use `wd:Q12418` for the Mona Lisa.
  - When merging, the node for Mona Lisa is unified.

5.7.2 Reification

- Used to make statements *about* other statements. [file:2]
- Example scenario: "Frankh created the statement about the location of the BaronWay Building". [file:2]
- Basic pattern:
  - Create a new resource representing the statement itself.
  - Attach properties for subject, predicate, object, creator, etc.
- This increases the number of triples (often 4 or more per original statement).

5.7.3 Named Graphs

- Alternative to full reification. [file:2]
- Assign a graph IRI to a set of triples.
- You can then describe the graph (e.g., its creator, provenance) using RDF about that graph IRI.

5.7.4 Modelling N-ary Relationships

- RDF predicates are binary (connect 2 resources). [file:2]
- For relationships involving 3+ entities (e.g., seller, buyer, broker of a sale), introduce an auxiliary resource.
- Pattern:
  - Create a node *SaleTransaction*.
  - Connect seller, buyer, broker to this node via separate properties.

6 Key Terminology Recap

From the recap table: [file:3]

- **Semantic web**: set of technologies (RDF, RDFS, OWL, IRI, SPARQL, SKOS, JSON-LD, etc.) to make data retrievable and processable by machines.
- **Web of data**: a web that implements the semantic web.
- **Open data**: data accessible online under a free licence.
- **Linked data**: data connected to other data using web/semantic technologies.
- **Linked open data**: open data that is also linked data.
- **Linked closed data**: linked data that is not openly licenced (enterprise internal data).
- **Knowledge graph**: knowledge base using a graph data model (concept from 1970s).
- **(Semantic) knowledge graph**: graph storing triples (subject, predicate, object).
- **Ontology**: semantic model of a domain (concepts, relationships, constraints).
- **Freebase**: knowledge graph (Metaweb, later Google) based on public datasets.
- **DBpedia**: early knowledge graph extracting structured data from Wikipedia. [file:3]

7 Exam-Oriented Summaries and Tips

7.1 Key Comparisons Table

Aspect	Traditional Web	Semantic Web
Main unit	HTML document	Resource / triple
Readable by	Humans	Humans and machines
Semantics	Implicit, in human mind	Explicit, machine-readable
Linking	Pages link to pages	Resources link to resources
Assumption	Mostly closed in DBs	Open world in web context
Data model	Tables, documents	Graph (RDF triples)

7.2 Typical Exam Questions You Can Expect

1. Define and differentiate.

- Data vs Information vs Knowledge.
- Web of documents vs Web of data.
- Open data vs Linked data vs Linked open data vs Linked closed data.

2. Explain technologies.

- What is RDF? What is a triple?
- What are IRIs and why are they important?
- Difference between RDF/XML, Turtle, N-Triples, JSON-LD, RDFa.
- What is OWL? What is an ontology?

3. Reasoning and modelling.

- Given a set of sentences (e.g., "John is a lecturer", "John teaches Big Data Management"), identify entities and relationships, then draw the knowledge graph. [file:3]
- Write RDF triples or Turtle for a given simple graph.
- Explain open world assumption with an example.

7.3 Minimal Formulas and Structures to Remember

RDF Triple Form.

$$\text{Triple} = (s, p, o)$$

- s: subject (IRI).
- p: predicate (IRI).
- o: object (IRI or literal).

Basic Turtle Pattern.

```
PREFIX ex: <http://example.org/>

ex:Bob  ex:knows   ex:Alice .
ex:Bob  ex:age     "30"^^xsd:int .
```

Open World Mindset.

- Not present in KB $\Rightarrow$ *unknown*, not automatically false.

7.4 High-Level Take-Home Message

- Semantic web = make data on the web *understandable* and *interoperable* for machines.
- Knowledge graphs = practical implementation: nodes (entities) and edges (relationships) capturing real-world facts.
- RDF + OWL + SPARQL = core tools to represent, model, and query this knowledge.