

Knowledge Graphs and Semantics for Question Answering

Acquisition, RDF/OWL representation, reasoning and a guided QA pipeline

FA-0769 · 4 days

Data & Analytics · Software Development

Introduction

Explore knowledge-graph concepts and apply them to a small question-answering system. Build an introductory pipeline that acquires knowledge, represents entities and relationships, queries a graph and evaluates answers.

The four-day programme combines conceptual discussion with prepared open-source tools and datasets. Advanced reasoning, machine-learning integration and hybrid systems are selected demonstrations and comparisons, not production deployment mastery.

Learning outcomes

- Explain knowledge-graph types, modelling choices and applications.
- Acquire, extract, normalise and evaluate domain knowledge.
- Develop a small schema or ontology and distinguish RDF, OWL and graph storage roles.
- Query a prepared RDF graph using SPARQL.
- Compare rule-based, semantic and machine-learning reasoning approaches.
- Build and evaluate a bounded question-answering example.
- Discuss hybrid systems, semantic-web relationships and open challenges.

Prerequisites

- Basic programming skills (preferably in Python)
- Familiarity with Natural Language Processing (NLP) concepts
- Familiarity with Database Systems and SQL

A compatible Python environment and prepared graph, extraction and reasoning tools; datasets must be authorised and suitable for the lab.

Training outline

1. Module 1 – Knowledge graph foundations

- Overview of knowledge graphs
- Types of knowledge graphs
- Knowledge graph modeling and representation
- Applications of knowledge graphs

Discuss the modelling decisions in a small domain example and compare graph types.

2. Module 2 – Knowledge acquisition

- Techniques for knowledge acquisition
- Knowledge extraction and normalization
- Domain-specific knowledge acquisition
- Quality control and evaluation of knowledge

Practise extraction and normalisation with prepared data; inspect source quality and domain assumptions.

3. Module 3 – Representation and querying

- Ontology and schema development
- RDF and OWL
- Graph databases and storage
- SPARQL query language

RDF expresses subject–predicate–object statements; OWL adds ontology semantics. Reasoning and query support depend on the chosen tool, not on every graph database automatically supporting every standard.

Use a small RDFLib/SPARQL exercise or equivalent prepared open-source environment.

4. Module 4 – Reasoning and inference

- Types of reasoning
- Rule-based reasoning
- Semantic inference
- Machine learning for knowledge graph-based QA

Compare inferred consequences with predicted answers. OWL open-world reasoning must not be confused with assuming every missing fact is false.

5. Module 5 – QA pipeline development

- QA System architecture
- Knowledge graph-based QA pipeline
- Entity and relation extraction
- Question parsing and answering
- Performance evaluation and optimization

Assemble and run a small prepared pipeline; inspect answer correctness, failure cases and measured performance before discussing optimisation.

6. Module 6 – Hybrid systems and open challenges

- Hybrid systems: combining traditional and knowledge graph-based QA
- Semantic web and knowledge graph-based QA
- Open challenges and future directions

Review the example and discuss current research directions using selected current sources; do not imply that an introductory course exhaustively covers the state of the art.