Ontology as the Defining Structure of a Digital Twin

Michael Grüninger

Semantic Technologies Lab
University of Toronto

Digital Twins in Finance 2024

Answering natural language questions about datasets

```
Alice supervises MATH456
MATH456 is a FinancialProject
Bob supervises MIE123
MIE123 is an EngineeringProject
```

- *Is Alice a manager in the division?*
- *Is Bob assigned to any projects?*

Ontologies

- What do “supervises”, “assigned to”, and “manager” mean?
- How do we represent this meaning and use it to answer questions?

An ontology is a declaration of terminology together with a specification of the meaning (semantics) of the terms within a formal logic.

Challenge

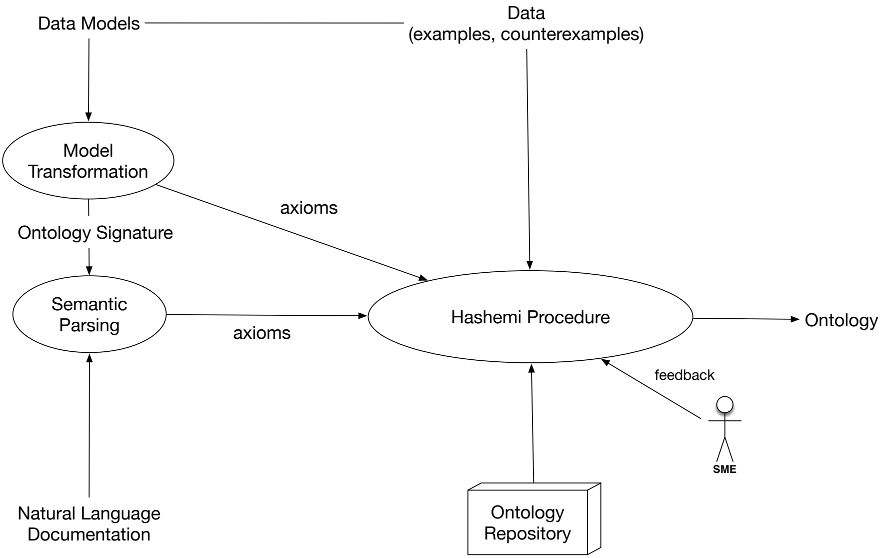
- Manually designing and evaluating an ontology is very resource-intensive, often requiring hundreds of work-hours.
- Even if we have an ontology, we still have the problem of integrating it with enterprise data.
- Using multiple ontologies often requires additional mappings to be manually specified.
- **Idea:** Rather than (manually) design the ontology first and then associate it with data, can we (semiautomatically) design the ontology from the data itself?

Overview

SEmi-Automated Design Of Ontologies (SEADOO)

- Generating modular ontologies through
  - data model transformation (*ontology grounding*)
  - transformation of natural language statements into logical sentences (*semantic parsing*)
  - learning from examples and user feedback (*Hashemi Procedure*)

SEADOO Architecture

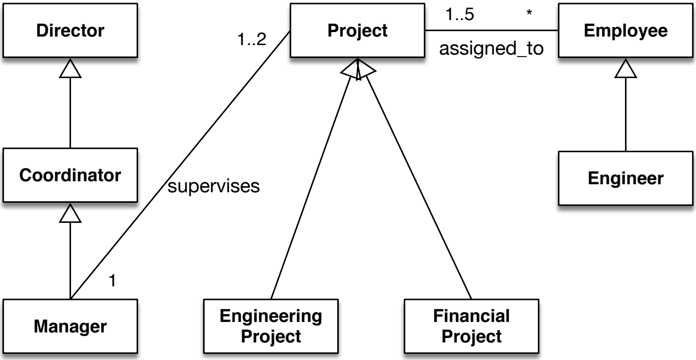


Ontologies and Data Models

Data models are seeds for ontologies:

- The classes and relations in the data model supply the signature for the ontology
- Constraints within the data model can be formalized by axioms of the ontology.

Data Model



Director
Project
1..5
*
Employee
1..2
assigned_to
Coordinator
supervises
Engineer
1
Manager
Engineering Project
Financial Project

Signature from the Data Model

Classes:

- $Manager(x)$, $Project(x)$, $Coordinator(x)$, $Supervisor(x)$,$Employee(x)$, $EngineeringProject(x)$, $FinancialProject(x)$

Relations:

- $supervises(x, y)$, $assigned\_to(x, y))$

Axioms from the Data Model

Managers supervise projects.

$$(\forall x, y)\ supervises(x, y) \supset Manager(x) \wedge Project(y) \tag{1}$$

Each manager supervises two different projects.

$$(\forall x) Manager(x) \supset (\exists y, z) supervises(x, y) \wedge supervises(x, z) \wedge (y \neq z) \tag{2}$$

Each project is supervised by a manager.

$$(\forall x)\ Project(x) \supset (\exists y)\ supervises(y, x) \wedge Manager(y) \tag{3}$$

Semantic Parsing

- The task of converting a natural language utterance to a logical form
  - Machine-readable representations: Prolog, lambda calculus, first-order logic, SQL, Python, Java, Abstract Meaning Representation
  - Used in machine translation, question answering, automated reasoning, and code generation

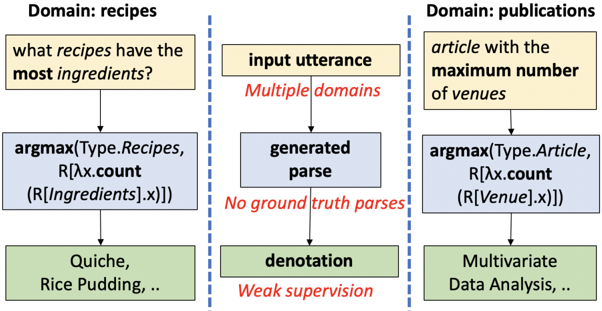


Figure: Image from [Agrawal2019]

Semantic Parsing: Current State

- The general problem of semantic parsing includes recognizing ontologies are needed for a given natural language sentence.
- We are currently considering a restricted problem:
  - We know the signature of the ontologies we are using
  - Sentences contain stative verbs.
- The first restriction allows us to specify a **phrase mapping**, which associates the classes and relations in the ontology signature with natural language phrases.
- Note: a demo video is available

Learning an Ontology

- In addition to ontology axioms that we can glean from data models and natural language documentation, can we learn ontologies directly from examples?

Intended Models

- Ontologies are designed with respect to a set of semantic requirements.
- There are various ways to specify requirements, such as competency questions and use cases.
- An ontology can be considered to be a set of logical theories whose purpose is to capture the intended interpretations corresponding to a certain conceptualization and to exclude the unintended interpretations.

Datasets as Intended Semantics

- Examples
  - Clean datasets without any semantic errors should be considered to be consistent with the ontology, and hence they can be considered to be a specification of intended models for the ontology.
- Counterexamples
  - Datasets with semantic quality problems should be inconsistent with the ontology. They can be considered to be a specification of interpretations that falsify one or more axioms of the ontology.

Hashemi Procedure: Discovery Phase

- Given a set of examples (intended models) and possibly a set of counterexamples (falsifying interpretations), search through the ontology repository to find the best match – an ontology whose models match all of the examples and none of the counterexamples.

Hashemi Procedure: Dialogue Phase

- Once a potential match has been found, generate a set of models and falsifying interpretations for the user.
- If the user agrees that the models are correct, then we conjecture that we have found the right ontology.
- If the user disagrees, then we refine the search.
- If we cannot find an exact match, then an ontology designer needs manually modify the best match to construct the ontology.

COLORE

- The COLORE (Common Logic Ontology Repository) project is building an open repository of ontologies specified using Common Logic (ISO 24707).
  - Testbed for ontology evaluation and integration techniques, and that can support the design, evaluation, and application of ontologies in first-order logic.

Seek and You Will Find?

- Since we are designing new ontologies, it will not be the case that the ontology we seek already exists, so how can the Hashemi Procedure possibly work?
- The answer can be found in ontology verification ...

Ontology Verification

- With verification, we want to characterize the models of an ontology up to isomorphism and determine whether or not these models are equivalent to the intended models.
- Relationships between first-order ontologies within a repository can be used to support ontology verification.
- The fundamental insight is that we can use the relationships between ontologies to assist us in the characterization of the models of the ontologies.
- The objective is the construction of the models of one ontology from the models of another ontology by exploiting the relationships between these ontologies and their modules in the repository.

COLORE Hypothesis

- *Every ontology is logically synonymous to the combination of mathematical theories that serve as ontology patterns in COLORE:*
  - Orderings
  - Graphs
  - Magmas
  - Mereographs
  - Subposets
  - Incidence structures
  - Geometries
  - Bundles and Foliations
  - Subgraph structures
  - Multigeometries

Translation Definitions

- In practice, users present their examples and counterexamples using the signature of their domain
  e.g. Project, Manager, supervises
- We use translation definitions to map the domain signature to the signatures of the mathematical theories.

$$(\forall x)\ Project(x) \equiv point(x) \tag{4}$$

$$(\forall x)\ Manager(x) \equiv line(x) \tag{5}$$

$$(\forall x, y)\ supervises(x, y) \equiv in(x, y) \wedge line(x) \wedge point(y) \tag{6}$$

Ontology Learning: Future Work

- Use the translation definitions to provide the user with a set of axioms that uses only the domain signature (the current implementation uses the combination of domain and mathematical signatures).
- The current implementation assumes that we have the right hierarchy within COLORE. How do we identify the right hierarchy from the datasets?
- How can we use the Hashemi Procedure to design a modularized ontology?

Summary

SEADOO is innovative – no commercial vendor we are aware of has the capabilities of SEADOO

Currently, we have developed the following components for SEADOO:

1. a tool to generate the axioms of an ontology from logical data models
2. a NLP model to extract NL sentences which are "axiom-worthy" from data models documentation
3. a semantic parser that can translate natural language questions into axioms
4. a tool (called Hashemi Procedure) which takes data (as examples/counter examples), mathematical theories (as ontology design patterns), initial axioms, and user feedback to generate ontologies