The IT Operations Ontology (ITOPS) resource

We have built a domain-agnostic pipeline for domain-specific ontologies that leverages information on Wikidata, Wikipedia and DBPedia. Each of the three stages of the pipeline extends the previous ones by using a variety of symbolic and ML/DL techniques. In this document we discuss the outcome of the pipeline in the IT Operations domain, the ITOPS ontology.

The product of each stage is a turtle (.ttl) file, ITOPS_S1.ttl, ITOPS_S2.ttl and ITOPS_S3.ttl, each extending the prior one. All of them build upon a General Library of Objects (GLO), which provides a general ontological framework to the domain specific graphs.

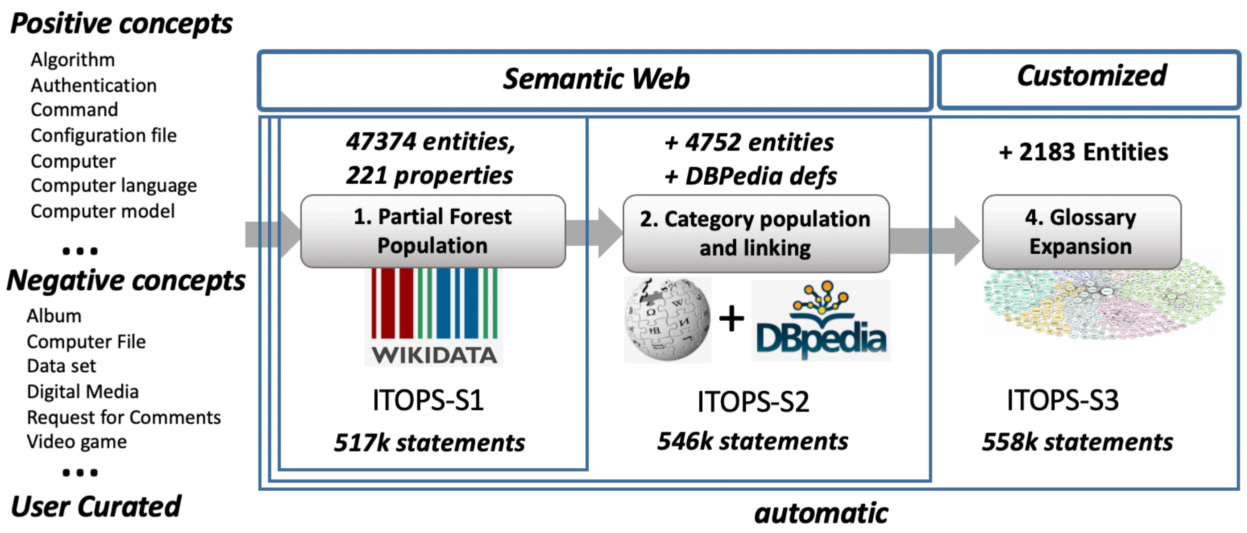
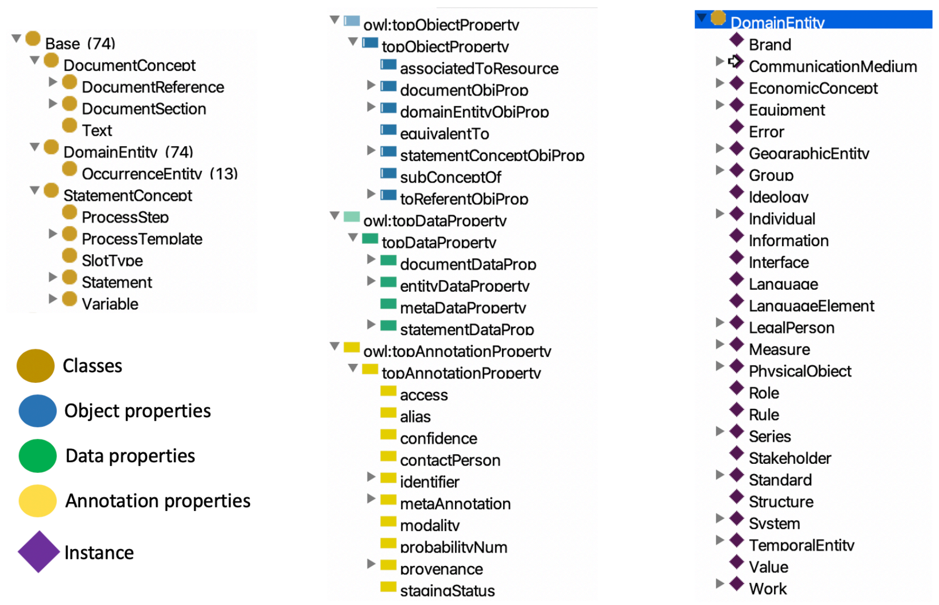


Figure 1. The pipeline used to build ITOPS

The General Library of Objects

We have created a small, general T-Box describing domain entities (glo:DomainEntity), statements about these entities and a set of high-level relations among them. We have populated it with 74 general-purpose entities derived from Wikidata. These entities are instances of glo:DomainEntity and are mostly related by a partial order relation, glo:subConceptOf, which subsumes wdt:P279 (subclass of) and wdt:P31 (instance of). This is done so that subsequent extensions to the ontology don't need to redraw the boundaries between T-Box and A-Box.

We call this T-Box plus the seed A-Box the General Library of Objects (GLO). A snapshot is depicted below.



Base (74)
- DocumentConcept
  - DocumentReference
  - DocumentSection
  - Text
- DomainEntity (74)
  - OccurrenceEntity (13)
- StatementConcept
  - ProcessStep
  - ProcessTemplate
  - SlotType
  - Statement
  - Variable

owl:topObjectProperty
- topObjectProperty
  - associatedToResource
  - documentObjProp
  - domainEntityObjProp
  - equivalentTo
  - statementConceptObjProp
  - subConceptOf
  - toReferentObjProp

owl:topDataProperty
- topDataProperty
  - documentDataProp
  - entityDataProperty
  - metaDataProperty
  - statementDataProp

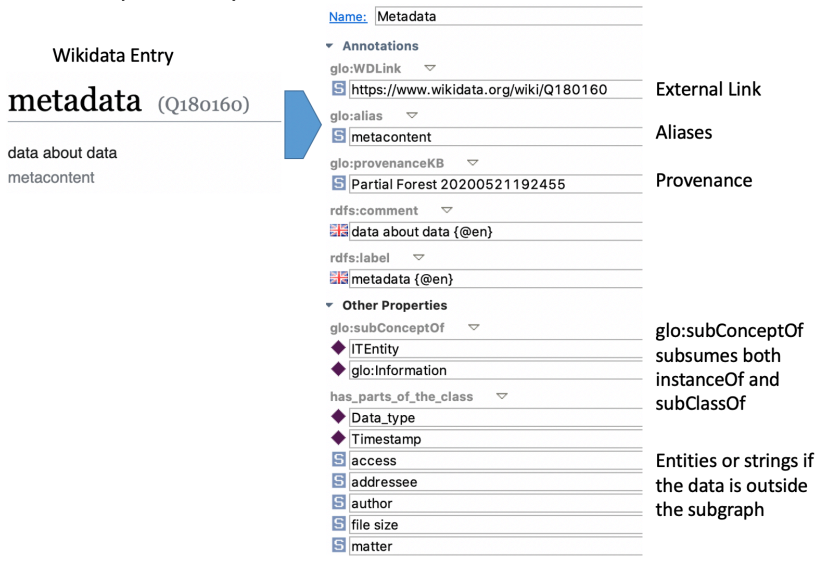
owl:topAnnotationProperty
- topAnnotationProperty
  - access
  - alias
  - confidence
  - contactPerson
  - identifier
  - metaAnnotation
  - modality
  - probabilityNum
  - provenance
  - stagingStatus

DomainEntity
- Brand
- CommunicationMedium
- EconomicConcept
- Equipment
- Error
- GeographicEntity
- Group
- Ideology
- Individual
- Information
- Interface
- Language
- LanguageElement
- LegalPerson
- Measure
- PhysicalObject
- Role
- Rule
- Series
- Stakeholder
- Standard
- Structure
- System
- TemporalEntity
- Value
- Work

● Classes
● Object properties
● Data properties
● Annotation properties
◆ Instance

Figure 2. GLO Upper Ontology

The GLO a shortcut for http://www.ibm.com/GLO# . In order to understand GLO's content better, let's look at a sample entity.


Wikidata Entry
External Link
Aliases
Provenance
glo:subConceptOf subsumes both instanceOf and subClassOf
Entities or strings if the data is outside the subgraph

metadata (Q180160)

data about data

metacontent

Name: Metadata

Annotations

glo:WDLink
https://www.wikidata.org/wiki/Q180160

glo:alias
metacontent

glo:provenanceKB
Partial Forest 20200521192455

rdfs:comment
data about data {@en}

rdfs:label
metadata {@en}

Other Properties

glo:subConceptOf
ITEntity
glo:Information

has_parts_of_the_class
Data_type
Timestamp
access
addressee
author
file size
matter

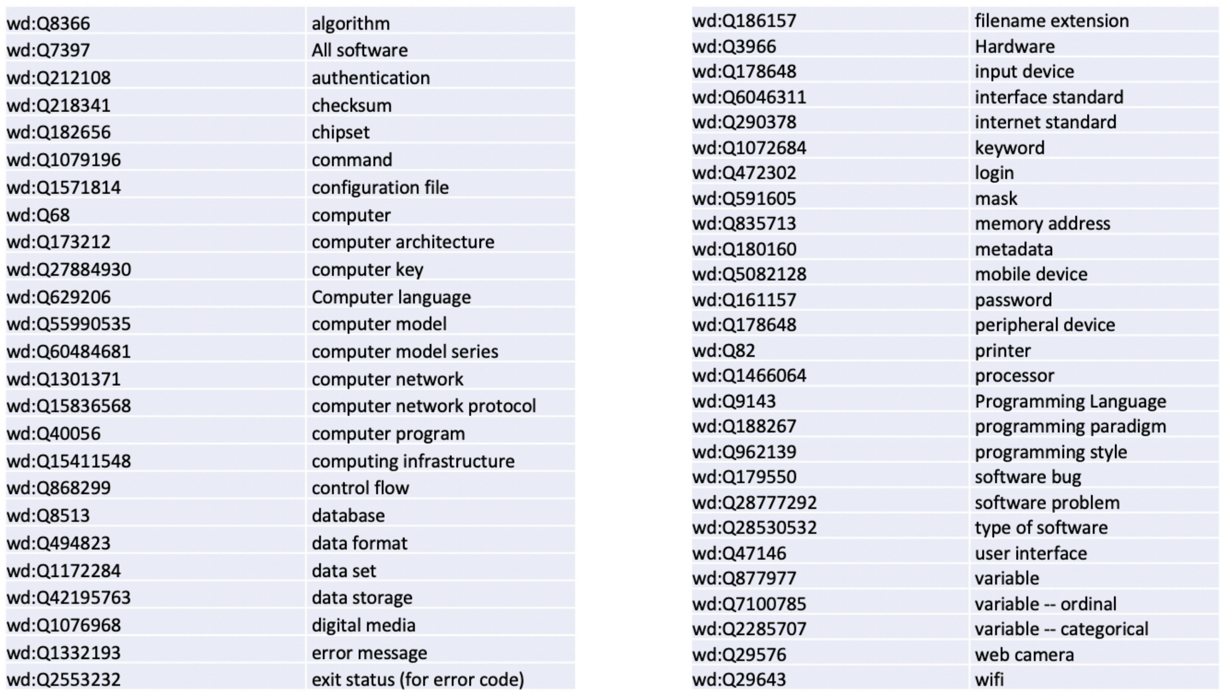
Figure 3. Sample entity using GLO vocabulary

Let's consider the entity "Metadata" in ITOPS, which uses the GLO. Each entity has metadata annotation properties associated to it, like aliases, comments, provenance (in this case indicating the step used to produce the entity) and external links (to Wikidata in this case). The relation glo:subConceptOf indicates that itops:Metadata is both an ITEntity and glo:Information. We discuss below how the inheritance hierarchy is constructed.

The relation itops:has_parts_of_the_class is one of the relations induced from the domain. These are defined as rdfs:subPropertyOf glo:domainEntityObjProp. All relations in a GLO ontology are categorized depending on whether they related domain entities (glo:domainEntityObjProp), statements (glo:statementConceptObjProp) or link statements to the domain entities they reference (glo:toReferentObjProp). Notice that some of the objects are instances, while others are strings. This is due to the construction pipeline, which seeks to limit the subgraph to relevant entities and avoids dangling references by referring to entities outside the scope of the ontology by strings.

S1 – The file ITOPS_S1.ttl

This file defines the itops namespace (http://www.ibm.com/ITOPS#) and is generated from a list of 52 common concepts in the TechQA dataset (https://arxiv.org/abs/1911.02984).



wd:Q8366	algorithm	wd:Q186157	filename extension
wd:Q7397	All software	wd:Q3966	Hardware
wd:Q212108	authentication	wd:Q178648	input device
wd:Q218341	checksum	wd:Q6046311	interface standard
wd:Q182656	chipset	wd:Q290378	internet standard
wd:Q1079196	command	wd:Q1072684	keyword
wd:Q1571814	configuration file	wd:Q472302	login
wd:Q68	computer	wd:Q591605	mask
wd:Q173212	computer architecture	wd:Q835713	memory address
wd:Q27884930	computer key	wd:Q180160	metadata
wd:Q629206	Computer language	wd:Q5082128	mobile device
wd:Q55990535	computer model	wd:Q161157	password
wd:Q60484681	computer model series	wd:Q178648	peripheral device
wd:Q1301371	computer network	wd:Q82	printer
wd:Q15836568	computer network protocol	wd:Q1466064	processor
wd:Q40056	computer program	wd:Q9143	Programming Language
wd:Q15411548	computing infrastructure	wd:Q188267	programming paradigm
wd:Q868299	control flow	wd:Q962139	programming style
wd:Q8513	database	wd:Q179550	software bug
wd:Q494823	data format	wd:Q28777292	software problem
wd:Q1172284	data set	wd:Q28530532	type of software
wd:Q42195763	data storage	wd:Q47146	user interface
wd:Q1076968	digital media	wd:Q877977	variable
wd:Q1332193	error message	wd:Q7100785	variable -- ordinal
wd:Q2553232	exit status (for error code)	wd:Q2285707	variable -- categorical
		wd:Q29576	web camera
		wd:Q29643	wifi

Figure 4. Seed concepts for S1

From these we generate the relevant subgraph from Wikidata and brought as children of itops:. The top level entities are then 'linked' to the GLO ontology by following the wdt:P31 and wdt:P279 relations. In Figure 3, itops:Metadata is a subConcept of both itops:ITEntity and glo:Information. Notice that the intermediate concepts (those between the top layer of ITOPS and the lower layers of GLO) are ignored at this time.

We call this process "Partial Forest" population. In the glo:provenanceKB of the items obtained like this, we include this term and a timestamp.

The result is a file with 47374 entities and 220 domain entity relation (plus subConceptOf, of course). In the last section we show a few sample queries to obtain topology information at the end of this document.

S2 – The file ITOPS_S2.ttl

The S2 version of ITOPS seeks to leverage the categories from the English Wikipedia. Most Wikipedia articles are manually labelled with tags corresponding to categories. Such tags can contain information that is not explicitly in Wikidata. Most entities in Wikidata can then be linked to their corresponding Wikipedia article through interwiki information. This allows us to get the associated Wikipedia categories for each Wikidata entity.

For instance, if we take an entity such as “computer keyboard” (Q250), the corresponding Wikipedia page is tagged with four categories “computer peripherals”, “computing input devices”, “flexible electronics”, and “game control methods”. While the first two are domain-specific categories containing mostly IT domain entities, the latter two are more generic. Thus, we need a method for distinguishing between those.

We have used metrics that indicate the heterogeneity of entities in a category and their overlap with the initial partial forest to identify domain-specific cate- gories. For that, we extract all Wikipedia articles tagged with a given category and collect their corresponding entities along with their class types. For exam- ple, the entities “computer peripherals” are typed as instances of few classes that subclasses of “peripheral equipment” (Q178648) class while entities in “flexible electronics” are typed from many diverse classes that are far from each other in the class hierarchy. Furthermore, a large portion (75%) of entities in ‘computer peripherals’ overlap with the initial partial forest while for ‘flexible electronics’ it is around 9%.

The result is a file with 51096 entities and 220 domain entity relation (plus subConceptOf, of course). In the last section we show a few sample queries to obtain topology information at the end of this document.

S3 – The file ITOPS_S3.ttl

S2 does not take care of customization of ITOPS ontology with domain specific resources. IT domain has glossaries readily available for multiple sub-domains. These glossaries capture specialized knowledge of a domain and are usually a good quality data source.

S3 uses Deep Learning to classify concepts from glossaries along the glo:subConceptOf hierarchy in the ontology.

Four such glossaries are used:

1. https://www.ibm.com/support/knowledgecenter/STXNRM_3.14.7/coss.doc/
2. https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.glossary.doc/doc/glossary.html
3. https://www.dpsolutions.com/success-center/it-terminology-glossary
4. https://flexsystem.lenovofiles.com/help

Due to IP concerns, only the name and external link to the terms are stored in ITOPS_S3.

The result is a file with 53193 entities and 220 domain entity relation (plus subConceptOf, of course). In the last section we show a few sample queries to obtain topology information at the end of this document.

QUERIES

These queries can be run either of the three files, as they all define the same namespaces.

QUERY: To obtain all properties in the graph (object, annotation, datatype)

```
prefix ITOPS: <http://www.ibm.com/ITOPS#>
prefix u2o: <http://www.ibm.com/U2O#>
SELECT DISTINCT ?property ?propertyLabel WHERE {
        {
                ?property rdfs:subPropertyOf u2o:topObjectProperty .
         } UNION
        {
                 ?property rdfs:subPropertyOf u2o:topDatatypeProperty .
        }
         UNION
        {
                 ?property rdfs:subPropertyOf u2o:topAnnotationProperty .
         }
```

QUERY: To obtain all entities in GLO

```
prefix ITOPS: <http://www.ibm.com/ITOPS#>
prefix u2o: <http://www.ibm.com/U2O#>
```

```
SELECT DISTINCT ?entity ?entityLabel WHERE {
            ?entity u2o:subConceptOf u2o:DomainEntity .
            FILTER(STRSTARTS(STR(?entity), "http://www.ibm.com/U2O#"))
            ?entity rdfs:label ?entityLabel .
 } ORDER BY ?entity
```

QUERY: to count all statements in the graph

```
SELECT DISTINCT (COUNT(*) as ?triples) WHERE
                        { ?s ?p ?o . }
```

QUERY: to count all statement minus those with metadata

```
prefix ITOPS: <http://www.ibm.com/ITOPS#>
prefix u2o: <http://www.ibm.com/U2O#>

SELECT DISTINCT (COUNT(*) as ?triples) WHERE {
    ?s ?p ?o .
   FILTER NOT EXISTS {
            ?p rdfs:subPropertyOf u2o:topAnnotationProperty .
            ?p rdfs:subPropertyOf u2o:topDataProperty .
             ?s rdfs:comment ?o .
            ?s rdfs:label ?o
     }
}
```