Text2Onto

A Framework for Ontology Learning and Data-driven Change Discovery

Philipp Cimiano, Johanna Völker

Pavel Shkadzko
University of Saarland
2015

Contents

Purpose of ontology learning

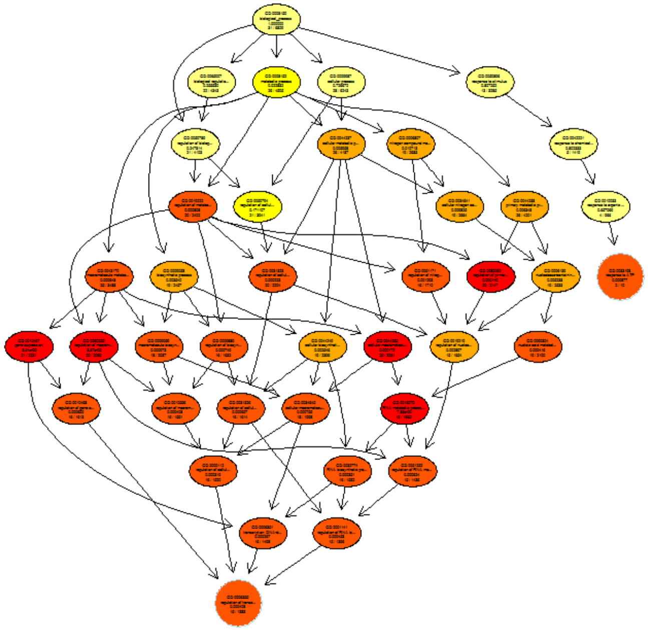
Problems of ontology learning

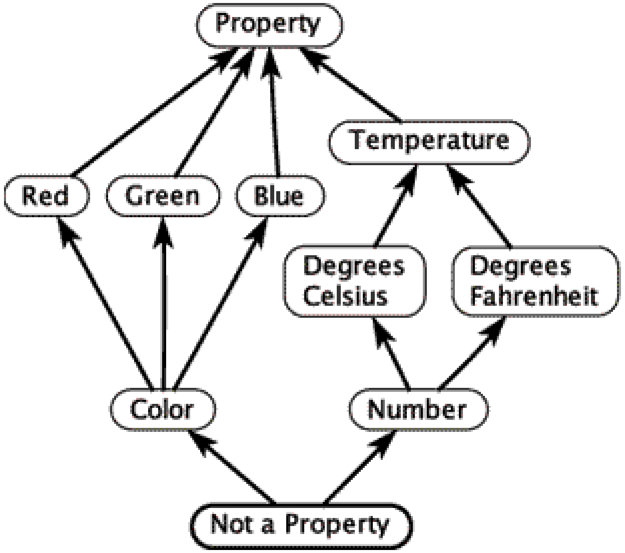
Text2Onto approach

Text2Onto architecture

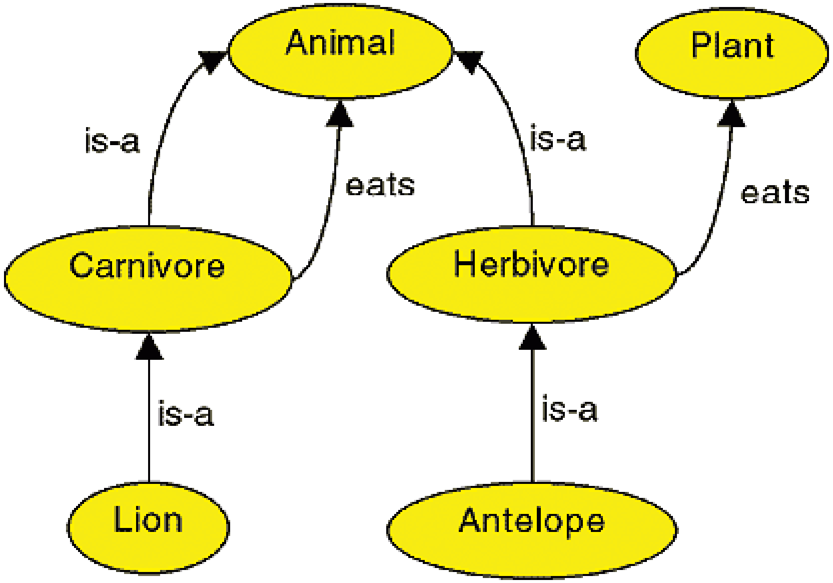
Performance analysis and results

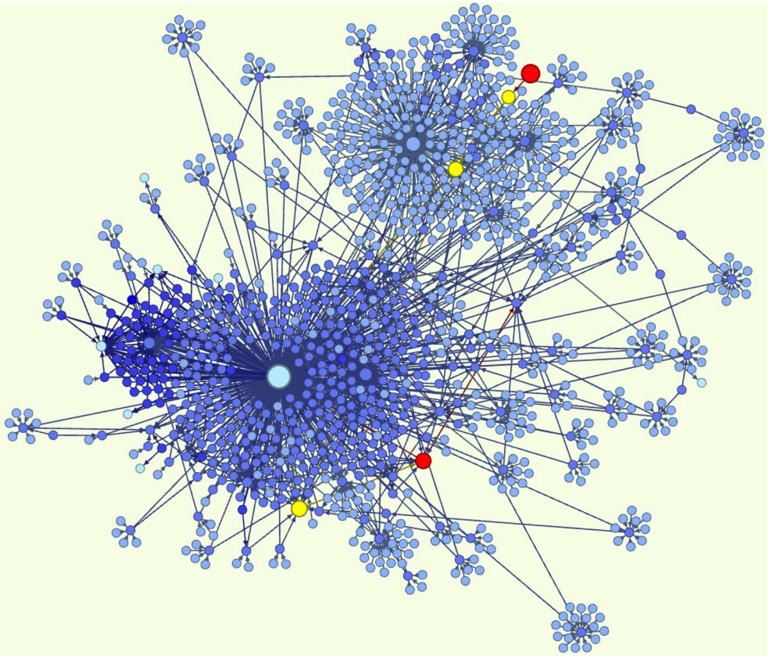
Questions



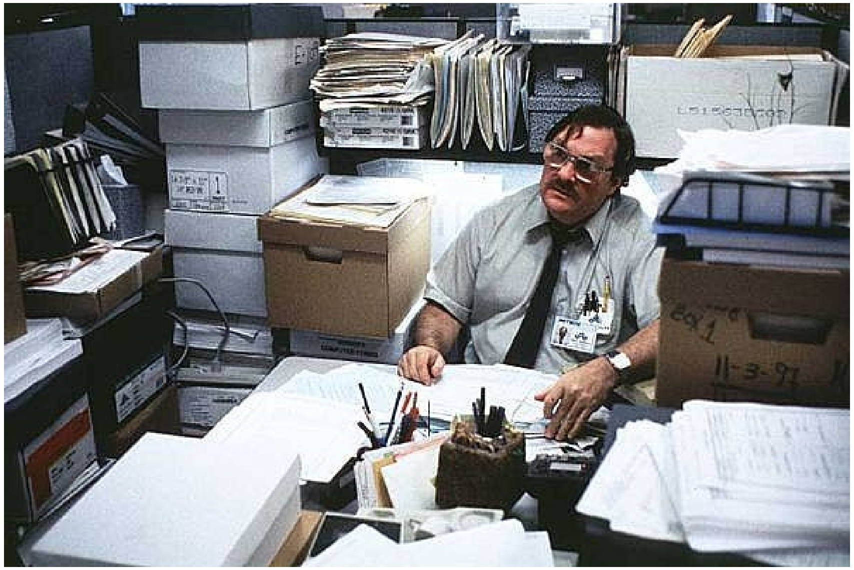


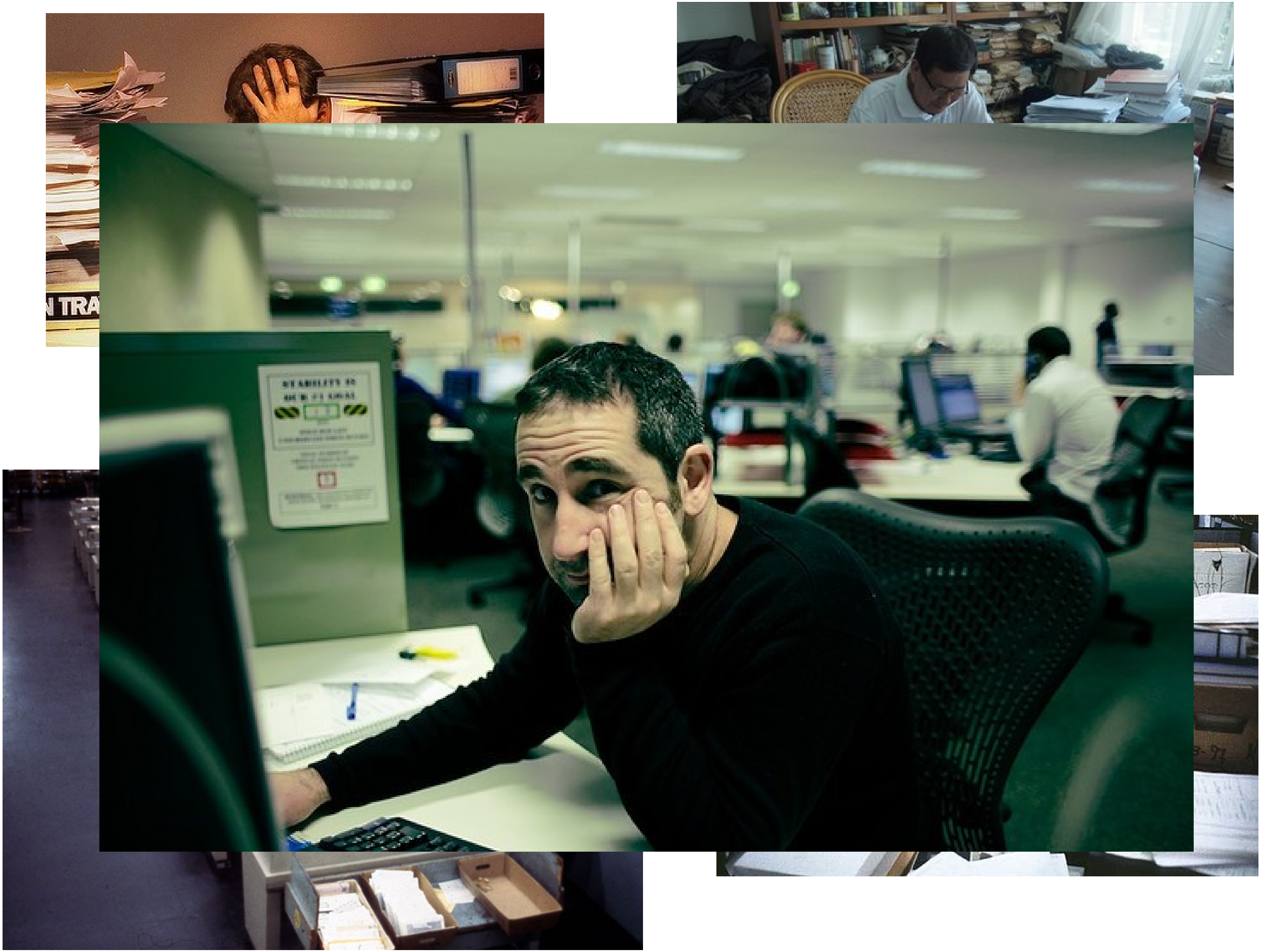
Systems for building and consolidating human knowledge





Manual ontology creation is expensive




N TRA

Is it that difficult?

Wikipedia says

“In artificial intelligence, an intelligent agent is an autonomous entity which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals.”

“A robot is a mechanical or virtual artificial agent, usually an electro-mechanical machine that is guided by a computer program or electronic circuitry.”

“R2-D2 is a robot character in the Star Wars universe created by George Lucas.”

Wikipedia says

*“In artificial intelligence, an **intelligent agent** is an **autonomous entity** which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals.”*

*“A **robot** is a mechanical or virtual **artificial agent**, usually an electro-mechanical machine that is guided by a computer program or electronic circuitry.”*

*“R2-D2 is a **robot** character in the Star Wars universe created by George Lucas.”*

Entity -> Autonomous entity -> Intelligent agent -> Robot

Wikipedia says

*“In **artificial intelligence**, an **intelligent agent** is an **autonomous entity** which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals.”*

*“A **robot** is a mechanical or virtual **artificial agent**, usually an **electro-mechanical machine** that is guided by a **computer program** or electronic **circuitry**.”*

*“**R2-D2** is a **robot** character in the **Star Wars** universe created by **George Lucas**.”*

?

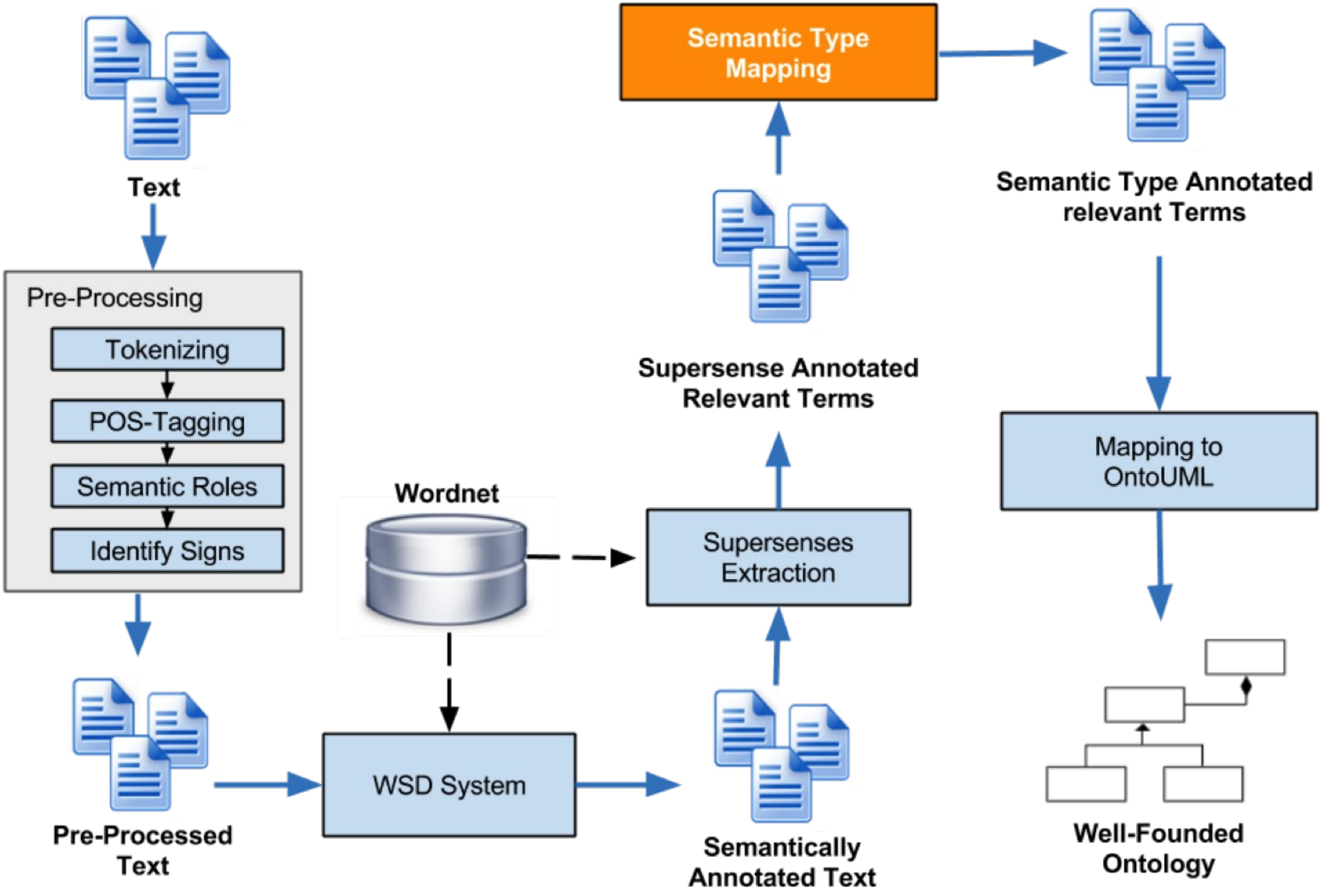
Wikipedia says

*“In **artificial intelligence**, an **intelligent agent** is an **autonomous entity** which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals.”*

*“A **robot** is a mechanical or virtual **artificial agent**, usually an **electro-mechanical machine** that is guided by a **computer program** or electronic **circuitry**.”*

*“**R2-D2** is a **robot** character in the **Star Wars** universe created by **George Lucas**.”*

artificial intelligence ? Star Wars ? intelligent agent ? machine ? George Lucas ? computer ? program ? robot ? R2-D2


Semantic Type Mapping
Semantic Type Annotated relevant Terms
Text
Pre-Processing
Tokenizing
POS-Tagging
Semantic Roles
Identify Signs
Supersense Annotated Relevant Terms
Mapping to OntoUML
Wordnet
Supersenses Extraction
WSD System
Pre-Processed Text
Semantically Annotated Text
Well-Founded Ontology

While it is quite easy for human beings to classify things the same task is overwhelming to a machine.

Automatic ontology creation poses a very serious challenge.

text2onto

Text2Onto is the official successor of TextToOnto, a framework for ontology learning from text.

The authors of Text2Onto addressed rather important issues that plagued many previous ontology learning frameworks.

1. Most ontology learning tools depend on specific or **ontology model** which chains them to one particular format (RDF, OWL, F-Logic).

The authors of Text2Onto addressed rather important issues that plagued many previous ontology learning frameworks.

1. Most ontology learning tools depend on specific or **ontology model** which chains them to one particular format (RDF, OWL, F-Logic).

2. **Interaction with end-users** is limited while user interaction should be the central part of the architecture.

The authors of Text2Onto addressed rather important issues that plagued many previous ontology learning frameworks.

1. Most ontology learning tools depend on specific or **ontology model** which chains them to one particular format (RDF, OWL, F-Logic).

2. **Interaction with end-users** is limited while user interaction should be the central part of the architecture.

3. Most state-of-the-art systems need to **recreate** ontology **from scratch** if corpus data has been modified.

How does Text2Onto solve the issues below?

Specific ontology model

User Interaction

Complete ontology rebuilding

How does Text2Onto solve the issues below?

Specific ontology model

User Interaction

Probabilistic Ontology Model (POM)

Complete ontology rebuilding

Data-driven Change Discovery

What is **Probabilistic Ontology Model (POM)**?

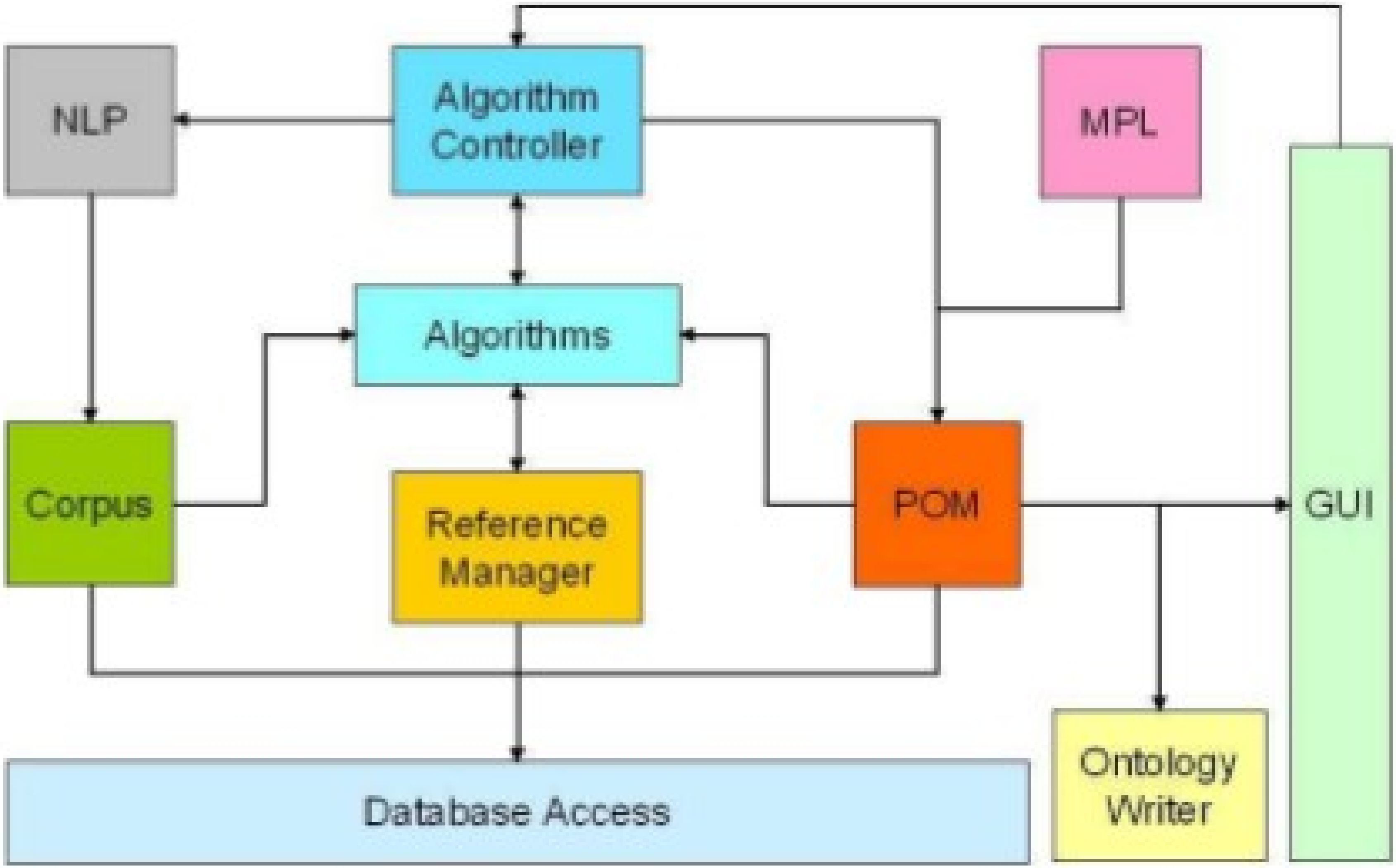
Probabilistic Ontology Model is a **container** for learned objects. All objects are enhanced by **calculated probabilities** in such manner that a user can decide whether to include this object into the ontology or not.

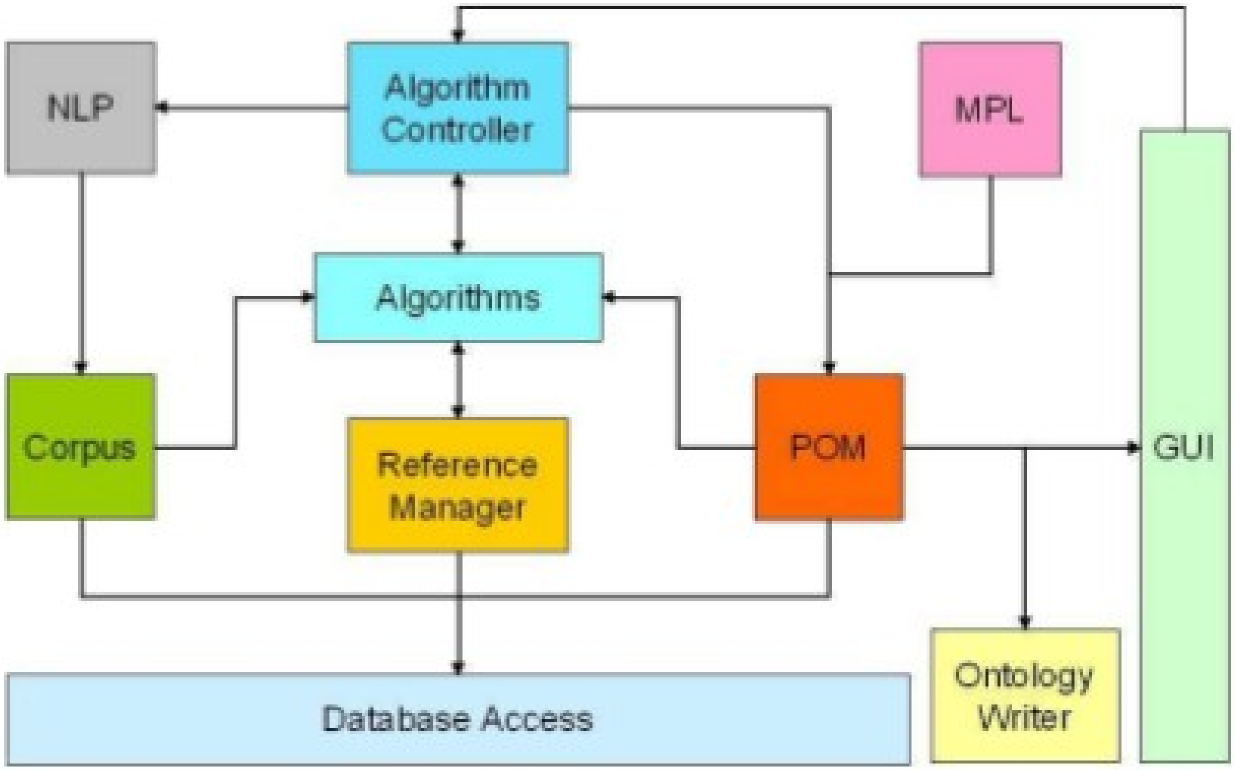
POM also stores a pointer for each object to whatever group of documents it is coming from.

What is **Data-driven Change Discovery**?

A **mechanism** that **tracks** the document **changes** and calculates POM data and corpus deltas making the changes to POM in return.

That allows for **in-place** ontology **updates** without triggering the whole process of rebuilding it from scratch.


NLP
Algorithm Controller
MPL
Algorithms
Corpus
Reference Manager
POM
GUI
Database Access
Ontology Writer



The main components of Text2Onto system are **NLP** engine, **Algorithms**, **Algorithm Controller** and **POM**.

Text2Onto NLP pipeline

Algorithm Controller triggers NLP pipeline.

NLP preprocessing step Text2Onto uses Gate4.0 framework for sentence detection, tokenization, and POS-tagging and application of JAPE pattern rules.

The results of **NLP preprocessing** are returned to **Algorithm Controller**.

What are JAPE rules?

JAPE -- jave annotation pattern engine.

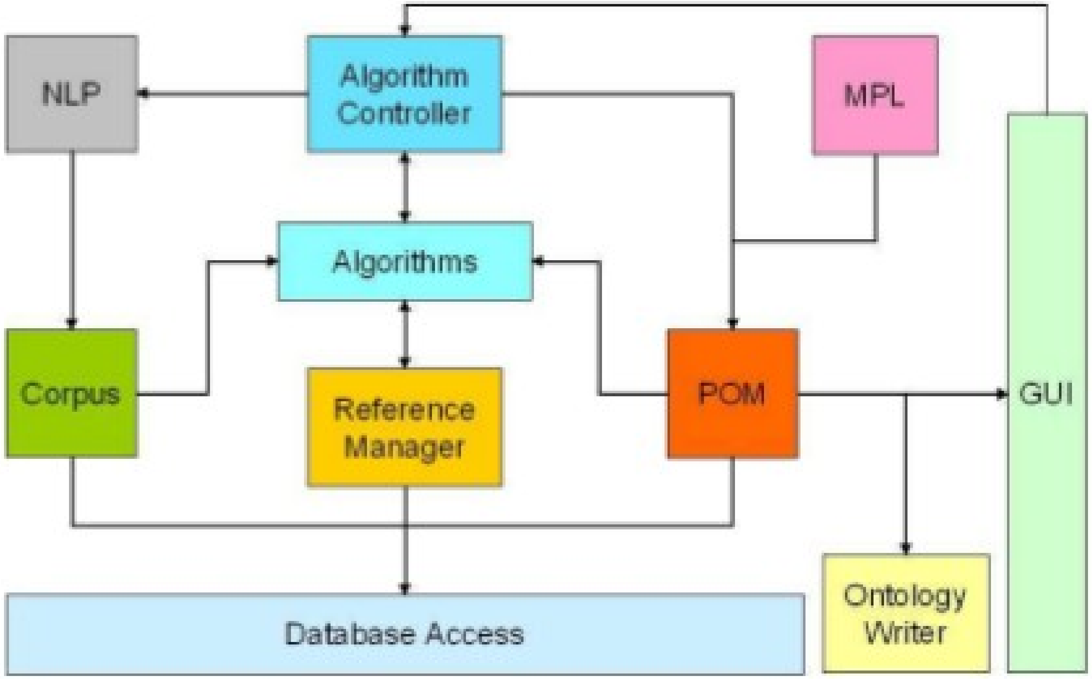
```
Rule: Jobtitle1
(
     {Lookup.majorType == jobtitle}
     (
         {Lookup.majorType == jobtitle}
     )?
)
:jobtitle
-->
:jobtitle.JobTitle = {rule = "JobTitle1"}
```

JAPE rules are language specific because of that Text2Onto supports only English, German and Spanish languages.

What are JAPE rules?

JAPE rule example

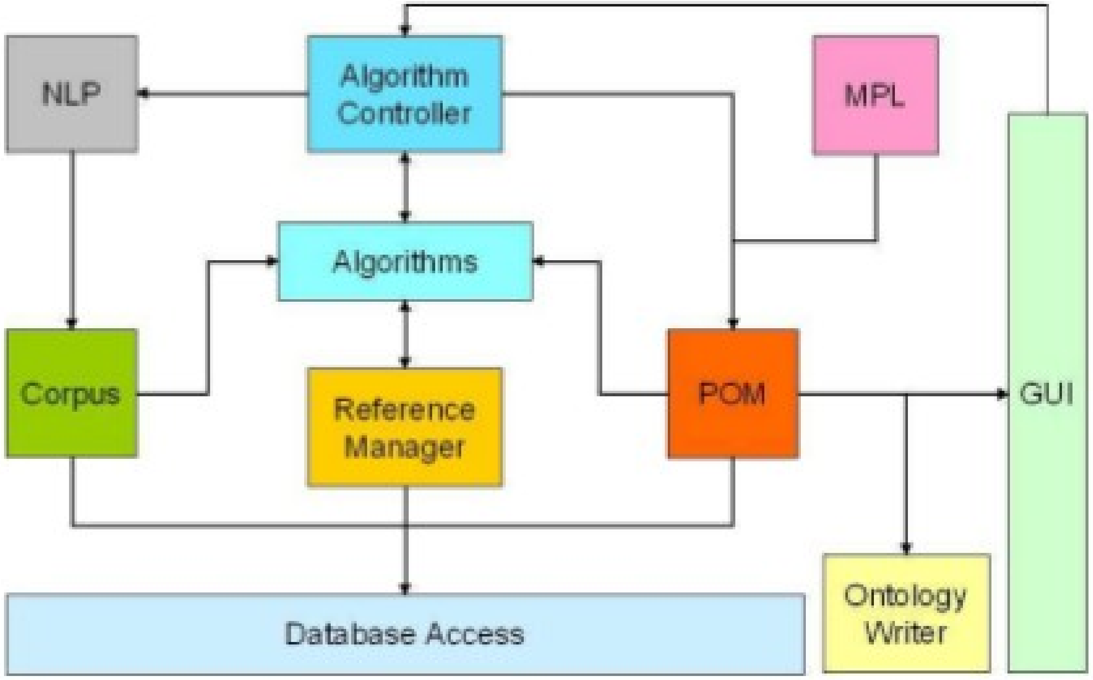
```
//professors or students
Rule: NP_Conj
(
   (NP):h (ConjCoor) (NP) ((ConjCoor) (NP))*
):phr
-->
:phr.Phrase = {^Ancestor =^ :h, Category="NP;COOR",
Semantic = "COOR"}
```



Algorithm Controller initializes **Algorithms** block.

Algorithms use the results returned by **NLP** component.
Recalculate probabilities and POM deltas.

POM stores the results.



Algorithm execution consists of 3 steps:

notification -- algorithm learns about changes in corpus if any,

calculation -- changes are mapped to the reference repository which stores connection information between data and ontology,

generation -- requests to make changes to POM are generated.

The algorithms can be combined in order to improve the confidence values.

What are POM primitives?

Special entities stored in POM.
POM primitives comply with Gruber's classification of ontology objects.

concepts -- *CLASS*
concept inheritance -- *SUBCLASS_OF*
concept instantiation -- *INSTANCE_OF*
relations -- *RELATION*
domain restrictions -- *DOMAIN*
mereological (part of) relations
equivalence

What kind of Algorithms Text2Onto uses?

For detecting important concepts:

Relative term frequency, **TF-IDF**, **Entropy, C** and **NC-value** methods. The values returned by the above algorithms are normalized and scaled between [0,1].

What kind of Algorithms Text2Onto uses?

Concept inheritance:

Wordnet is used and its **hypernym** network.

Mereological relations:

JAPE rules are used + **Wordnet**

General relations:

JAPE rules are used.

Not all relations are detected, only:

transitive -- **love(Tom, Jerry)**,

intransitive + *PP* -- **chase(Tom, pp(until))**,

transitive + *PP* -- **chase(Tom, Jerry, pp(until))**,

What kind of Algorithms Text2Onto uses?

Concept Instantiation:

Vector similarity based approach, where instance and concept vectors are extracted from corpus and instance vectors closest to concept vectors are assigned to such concepts.

Equivalence:

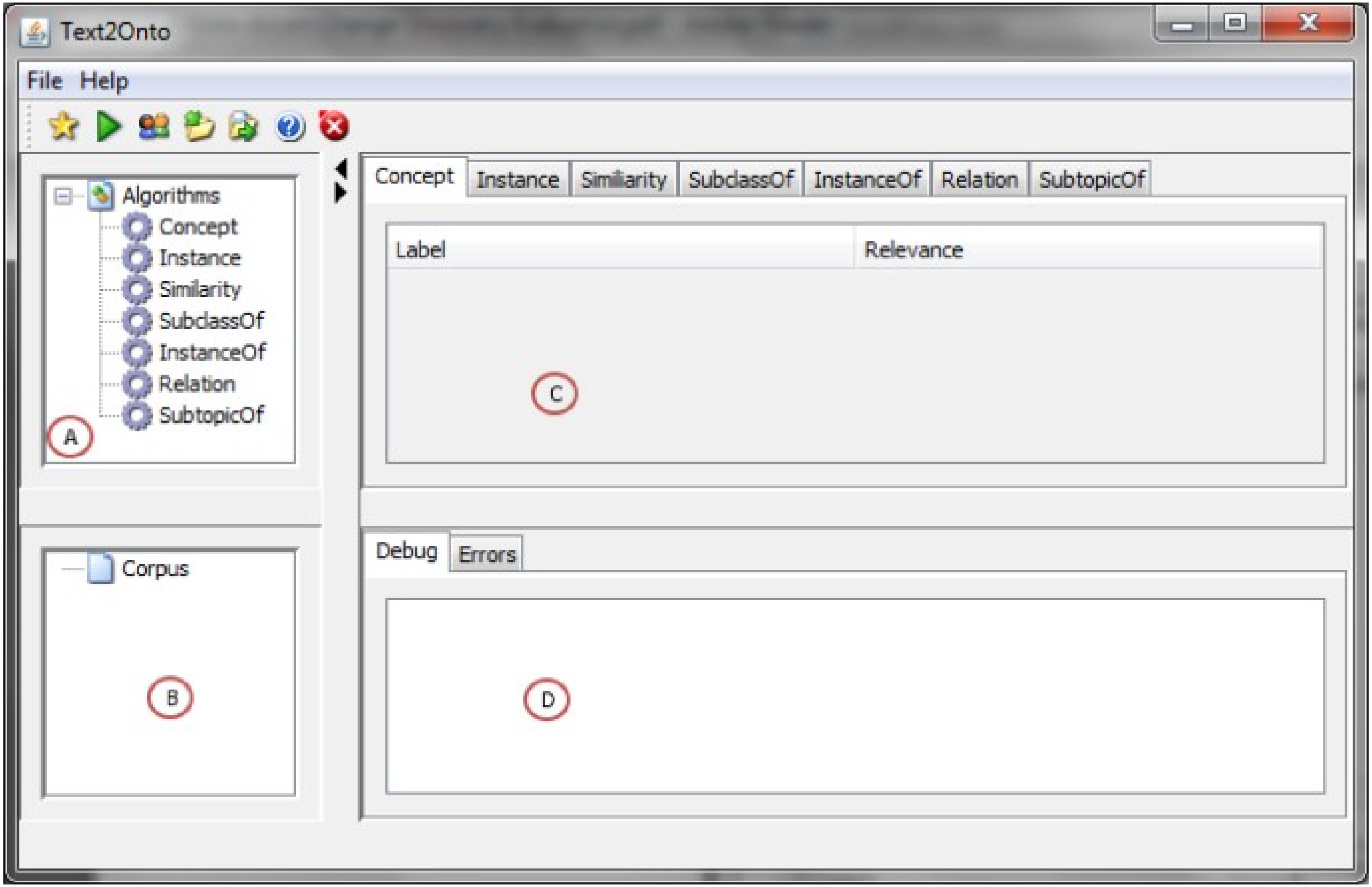
Context similarity between terms and concepts is calculated.

What kind of Algorithms Text2Onto uses?

The results of different algorithms are then combined via **predefined combination strategies**.

Each algorithm is provided with a **reference store** where its results can be saved and accessed later.

Text2Onto Workflow

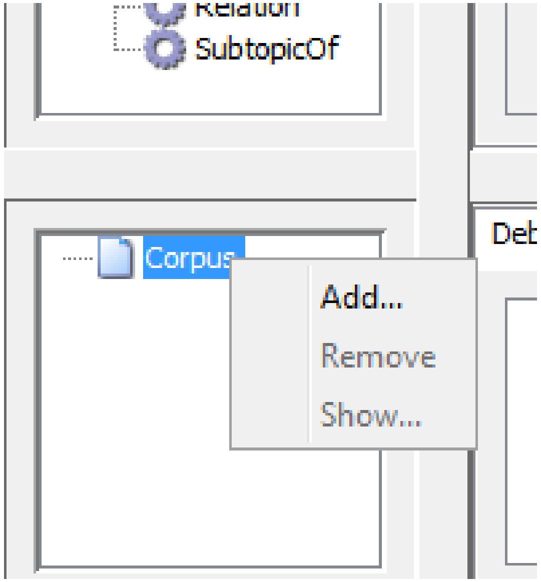


A. Controller view where we specify which Algorithms to use and how to combine the results of these algorithms.

B. Corpus view from where adding / removing a corpus is done.

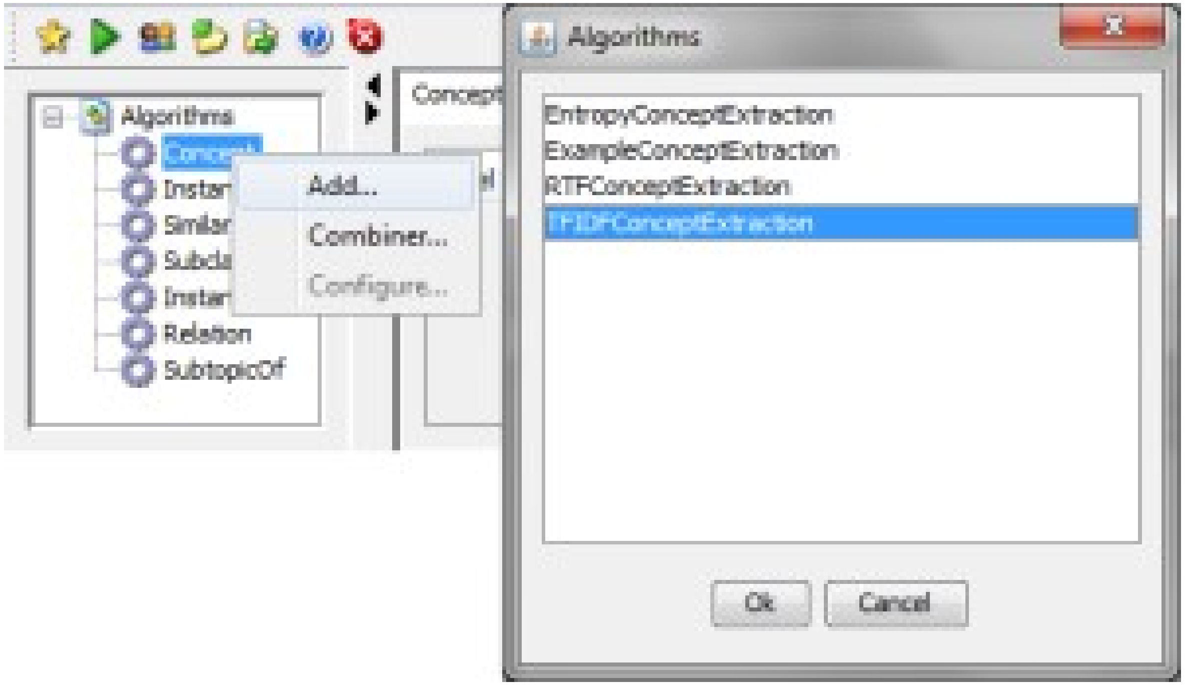
C. POM view panel. Displays the results of the current ontology learning process.

D. Displays debugging messages and error messages.



Step 1: **Add a Corpus**

Right-click on the label ‘Corpus’ on corpus view panel (B) and add a corpus.



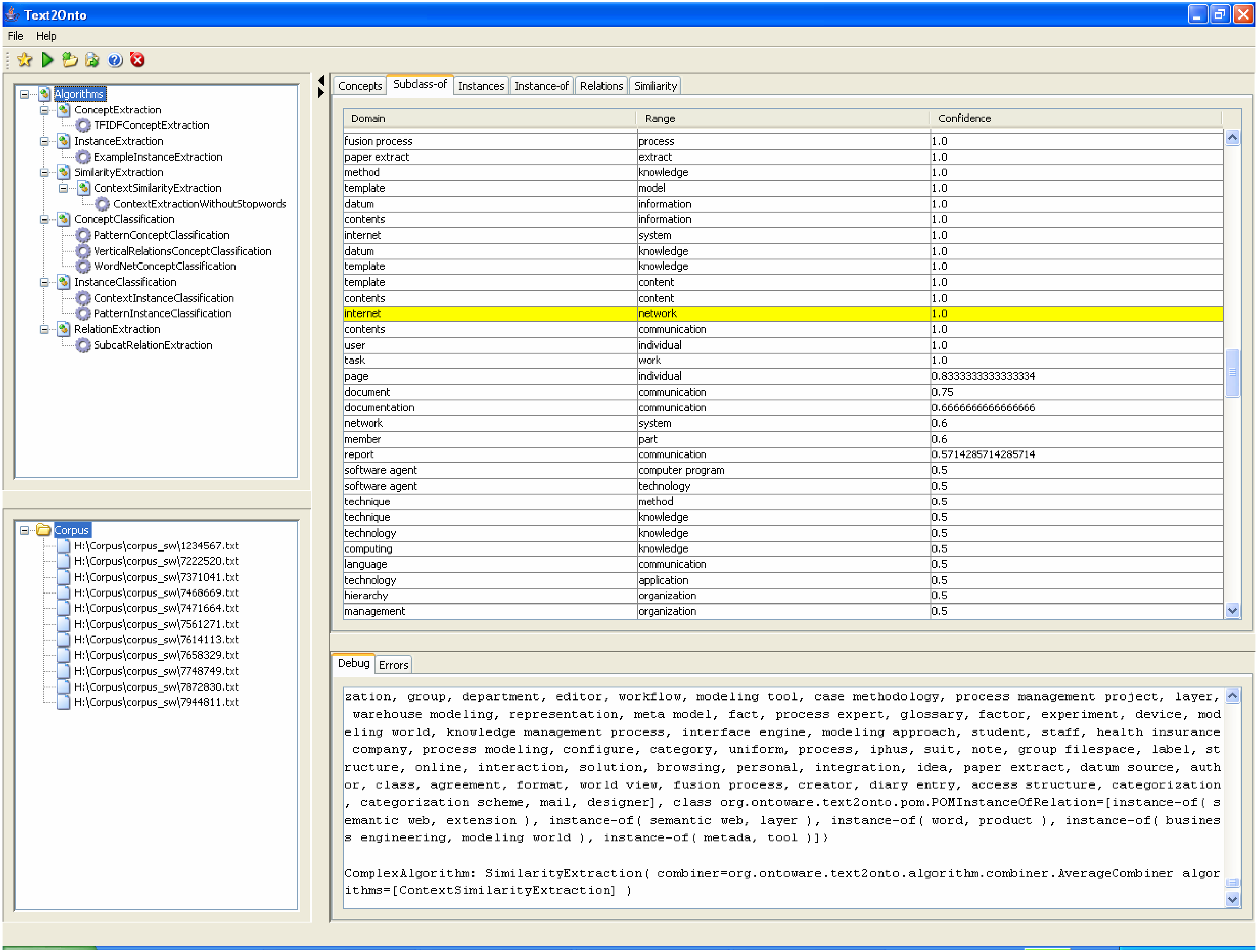
Step 2: **Specify algorithms to be applied**

Right-click on the required entity on the controller view panel (A) and click add. A list of available algorithms will appear.

You can add one or more algorithms from here.

Step 3: **Run**

Once all required algorithms have been specified, click the ‘Run’ icon (the second icon on the toolbar) to execute the process. The results will appear on the POM view panel (C).



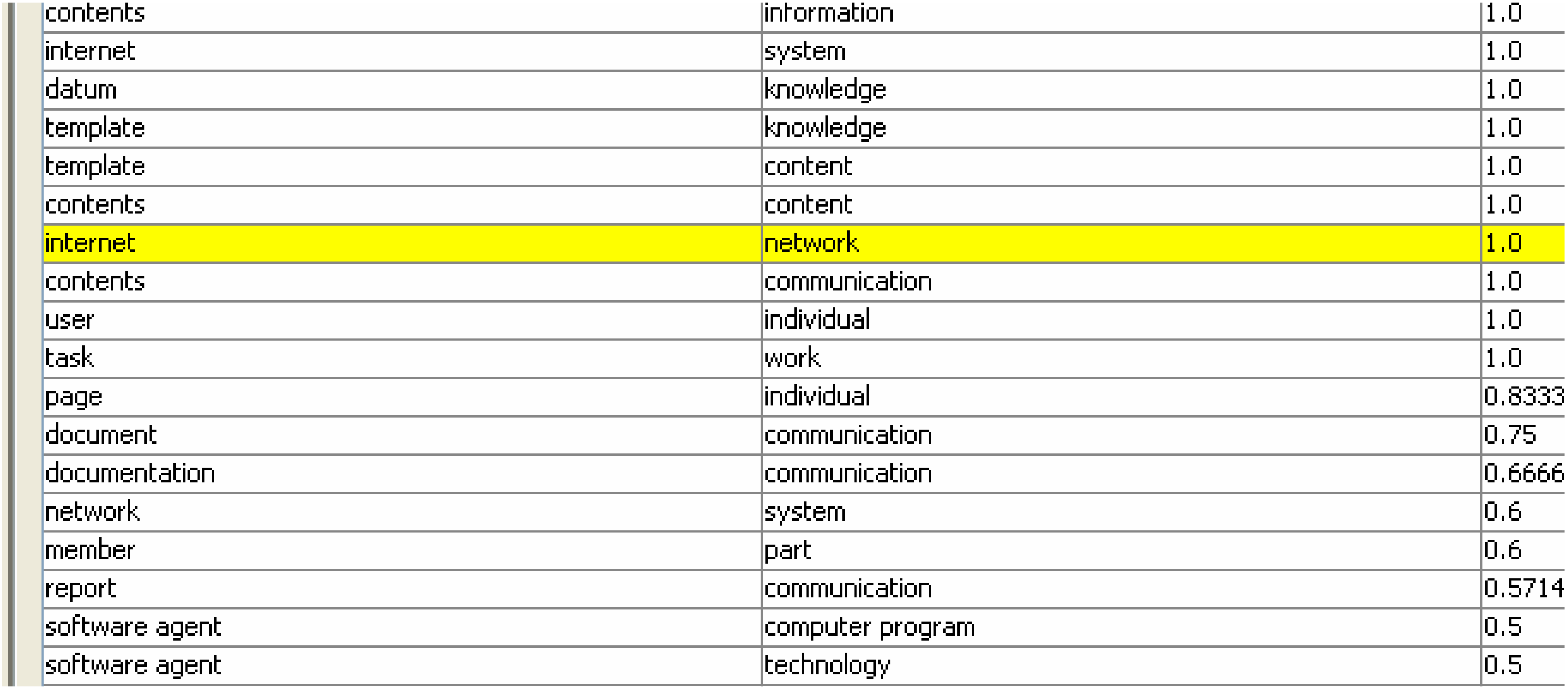
- Algorithms
  - ConceptExtraction
    - TFIDFConceptExtraction
  - InstanceExtraction
    - ExampleInstanceExtraction
  - SimilarityExtraction
    - ContextSimilarityExtraction
      - ContextExtractionWithoutStopwords
  - ConceptClassification
    - PatternConceptClassification
    - VerticalRelationsConceptClassification
    - WordNetConceptClassification
  - InstanceClassification
    - ContextInstanceClassification
    - PatternInstanceClassification
  - RelationExtraction
    - SubcatRelationExtraction

- Corpus
  - H:\Corpus\corpus_sw\1234567.txt
  - H:\Corpus\corpus_sw\7222520.txt
  - H:\Corpus\corpus_sw\7371041.txt
  - H:\Corpus\corpus_sw\7468669.txt
  - H:\Corpus\corpus_sw\7471664.txt
  - H:\Corpus\corpus_sw\7561271.txt
  - H:\Corpus\corpus_sw\7614113.txt
  - H:\Corpus\corpus_sw\7658329.txt
  - H:\Corpus\corpus_sw\7748749.txt
  - H:\Corpus\corpus_sw\7872830.txt
  - H:\Corpus\corpus_sw\7944811.txt

Domain	Range	Confidence
fusion process	process	1.0
paper extract	extract	1.0
method	knowledge	1.0
template	model	1.0
datum	information	1.0
contents	information	1.0
internet	system	1.0
datum	knowledge	1.0
template	knowledge	1.0
template	content	1.0
contents	content	1.0
internet	network	1.0
contents	communication	1.0
user	individual	1.0
task	work	1.0
page	individual	0.8333333333333334
document	communication	0.75
documentation	communication	0.6666666666666666
network	system	0.6
member	part	0.6
report	communication	0.5714285714285714
software agent	computer program	0.5
software agent	technology	0.5
technique	method	0.5
technique	knowledge	0.5
technology	knowledge	0.5
computing	knowledge	0.5
language	communication	0.5
technology	application	0.5
hierarchy	organization	0.5
management	organization	0.5

```
zation, group, department, editor, workflow, modeling tool, case methodology, process management project, layer,
 warehouse modeling, representation, meta model, fact, process expert, glossary, factor, experiment, device, mod
eling world, knowledge management process, interface engine, modeling approach, student, staff, health insurance
 company, process modeling, configure, category, uniform, process, iphus, suit, note, group filespace, label, st
ructure, online, interaction, solution, browsing, personal, integration, idea, paper extract, datum source, auth
or, class, agreement, format, world view, fusion process, creator, diary entry, access structure, categorization
, categorization scheme, mail, designer], class org.ontoware.text2onto.pom.POMInstanceOfRelation=[instance-of( s
emantic web, extension ), instance-of( semantic web, layer ), instance-of( word, product ), instance-of( busines
s engineering, modeling world ), instance-of( metada, tool )]]

ComplexAlgorithm: SimilarityExtraction( combiner=org.ontoware.text2onto.algorithm.combiner.AverageCombiner algor
ithms=[ContextSimilarityExtraction] )
```



contents	information	1.0
internet	system	1.0
datum	knowledge	1.0
template	knowledge	1.0
template	content	1.0
contents	content	1.0
internet	network	1.0
contents	communication	1.0
user	individual	1.0
task	work	1.0
page	individual	0.8333
document	communication	0.75
documentation	communication	0.6666
network	system	0.6
member	part	0.6
report	communication	0.5714
software agent	computer program	0.5
software agent	technology	0.5

Step 4: **Review the results**

The results of Text2Onto may need to be filtered. We can do this by giving feedback to it. To give feedback, right-click on the required entity, go to feedback and set the appropriate feedback (True, False or Don't know).

Step 5: **Export the results**

The results from Text2Onto can be exported in KAON, RDFS or OWL format. To do this, go to File and click Export. There is also a provision for saving the current session. However, it was not working at the time of this writing.

Performance Analysis

Best result based on tourism related texts:

F-Measure -- 21.81%

Precision -- 17.38%

Recall -- 29.95%

Can Text2Onto **automatically build** an ontology by **learning** on a corpus of texts?

Can Text2Onto **automatically build** an ontology by **learning** on a corpus of texts?

No

Can Text2Onto **automatically build** an ontology by **learning** on a corpus of texts?

No

Can Text2Onto **help** a user to build an ontology?

Can Text2Onto **automatically build** an ontology by **learning** on a corpus of texts?

No

Can Text2Onto **help** a user to build an ontology?

Yes, but it needs improvement

Thank you