

Ontology Learning

Lama Saeeda

saeeda.lama@fel.cvut.cz

December 17, 2020



Overview

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion

Ontology and Ontology Learning



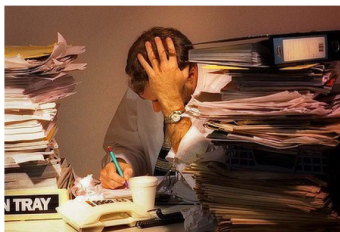
Basics

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



- Applications with ontology
- Jim Hendler: “a little semantic goes a long way”
- Availability, suitability, completeness





Manual ontology creation is expensive



Ontology Definition

- 1 **Ontology and Ontology Learning**
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



What is ontology?

- "Specification of a conceptualization" Tom Gruber
- "A description of things that exist and how they relate to each other"
Chris Welty



Ontology components

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



What are the components of the ontology?

Ontology can be defined as a tuple:

$$\vartheta = (C, R, H^C, rel, A^\vartheta)$$

- C is the set of ontology concepts. The concepts represent the entities of the domain being modeled. They are designated by one or more natural language terms and are normally referenced inside the ontology by a unique identifier.
- $H^C \subseteq C \times C$ is a set of taxonomic relationships between the concepts. Such relationships define the concept hierarchy.
- R is the set of non-taxonomic relationships. The function $rel : R \rightarrow C \times C$ maps the relation identifiers to the actual relationships.
- A^ϑ is a set of axioms, usually formalized into some logic language. These axioms specify additional constraints on the ontology and can be used in ontology consistency checking and for inferring new knowledge from the ontology through some inference mechanism.



Ontology Learning

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion

Methods of Ontology Learning



Layer-cake model

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Layer-cake model for learning ontology

$\forall x, y (sufferFrom(x, y) \rightarrow ill(x))$

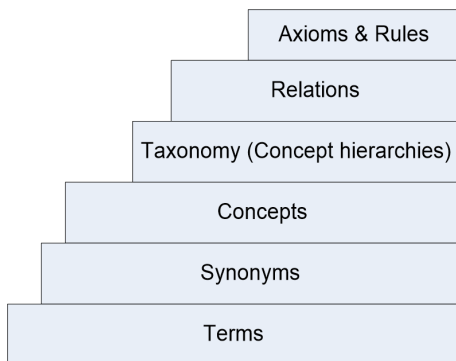
cure (domain:Doctor, range:Disease)

is_a (Doctor, Person)

Disease := <I, E, L>

{disease, illness}

disease, illness, hospital



input

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Possible input sources

- Structured data - database schemes
- Semi-structured data - dictionaries like **WordNet**
- Unstructured data - natural language text documents, like the majority of the HTML based web-pages



- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Learning methods

- Linguistic
- Statistical
- Rule-Based
- Logical

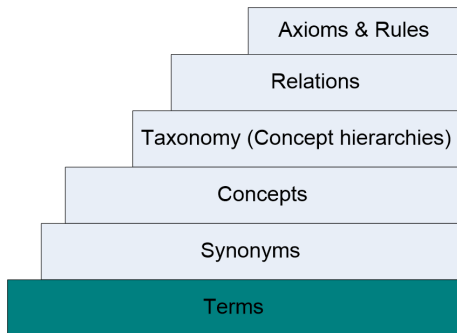


Learning terms

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Terms extraction



disease, illness, hospital



Terms extraction - Linguistic processing

Natural Language Processing (NLP), deep language analysis or information retrieval methods for term indexing.

- **Identifies** sentences, determined by periods or other punctuation marks
- **Tokenization** separates text into tokens which are the basic units
- **Normalizes** tokens to lower case to provide case-insensitive indexing
- **Stemming**: (fishing, fished, fisher) one stem: **fish**
- **Stop-words removing**: Meaningless tokens, (**there, so, other, etc..**)
- **POS tagging**: the **book** on the table (noun), to **book** a flight (verb)



Terms extraction - Statistical metrics

- **TF: Term Frequency**, how frequently a term occurs in **one document**.
TF = (Number of times term t appears in a document / Total number of terms in the document)
- **IDF: Inverse Document Frequency**, how important a term is in the **corpus** IDF = \log (Total number of documents / Number of documents with term t in it)



Terms extraction - Statistical metrics

$$tfidf(w) = tf(w) \cdot \log\left(\frac{N}{df(w)}\right)$$

The word is more popular when it appears several times in a document

The word is more important if it appears in less documents

- $tf(w)$ → term frequency (number of words occurrences in a document)
- $df(w)$ → document frequency (number of documents containing the word)
- N → number of all documents
- $tfidf$ → relative importance of the word in the document



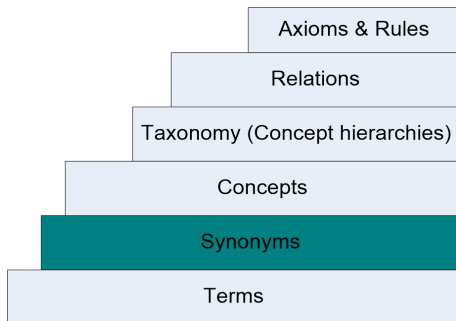
Learning synonyms

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Synonyms extraction

{disease, illness}



Synonyms extraction

Identification of terms that share semantics, i.e., potentially refer to the same concept

- Latent Semantic Indexing (LSI): Assumes that words that are close in meaning will occur in similar pieces of text (the distributional hypothesis).
- Wordnet



Synonyms extraction - Wordnet overview

What is wordnet?

- General lexical knowledge base
- Contains 150,000 words (noun, verb, adj, adv)
- A word can have multiple senses: “plant” as a noun has 4 senses
- Each concept (under each sense and PoS) is represented by a set of synonyms (a syn-set).
- Semantic relations such as hypernym/antonym/meronym of a syn-set are represented



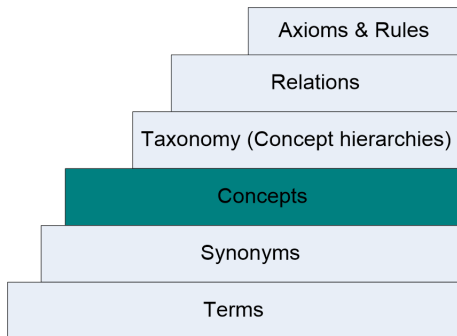
Learning concepts

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Concepts extraction

Disease := $\langle I, E, L \rangle$



Concepts

Controversial as it is not clear what exactly constitutes a concept

A term may indicate a concept, if we define its:

- **Intension** (In)formal definition of the objects this concept describes
ex: a disease is an impairment of health or a condition of abnormal functioning
- **Extension** Set of objects described by this concept (ontology population)
ex: influenza, cancer, heart disease
- **Lexical Realizations** The term itself and its multilingual synonyms
ex: disease, illness, maladie



Concepts forming approaches

The detection of synonyms can help to cluster terms to groups of terms sharing (almost) the same meaning, thus representing ontological classes.

- Learning the extension of concepts
 - Unsupervised hierarchical clustering techniques known from machine learning research
 - Clusters of related terms (overlaps almost completely with *term* and *synonym* extraction)
 - **for example** "all movie actors appearing on the Web"
- Learning the intension of concepts
 - Acquisition of informal definition
 - Textual description, i.e. a gloss of the concept (ex. from Wordnet), normative document, etc.
 - Acquisition of formal definition
 - Discovering formal constraints that define a class membership



Concepts labeling

Hearst's patterns

- 1 Find hypernym candidates for each class members
- 2 Then select the top candidate related to the largest number of class members

Web search

- 1 Proper query of concatenating the child concepts
- 2 Return top 10 results + NLP
- 3 Select the most frequent phrase

- Challenging problem
- Assign meaningful name to the newly-created parent node
- General enough to cover the scope of all the child concepts
- Specific enough to just cover that of them

Ex: *president* becomes the parent of concepts *Bush* and *Reagan*.



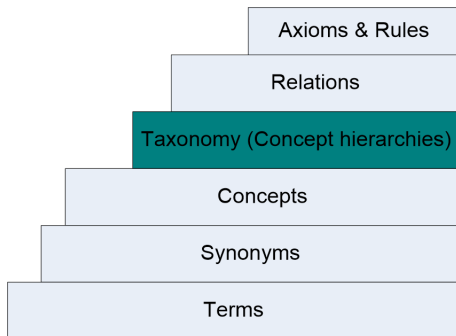
Learning concept hierarchies

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Taxonomy

is_a (Doctor, Person)



Learning concept hierarchies

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Taxonomy

“is-a” hierarchy on concepts

Existing approaches

- 1 Hearst Patterns (Lexico-syntactic patterns)
- 2 Hierarchical Clustering
- 3 Document-based subsumption



Taxonomy - Hearst Patterns

- The acquisition of hyponym lexical relations from text
- Uses a set of predefined lexico-syntactic patterns which:

- Occur frequently and in many text genres
- Indicate the relation of interest
- Can be recognized with little or no pre-encoded knowledge

- Principle idea: match these patterns in texts to retrieve is-a relations
- Reasonable *precision*, very low *recall*



Taxonomy - Hearst Patterns

- Vehicles **such as** cars, trucks and bikes
- **Such** fruits **as** oranges or apples
- Swimming, running **and other** activities
- Swimming, running **or other** activities
- Publications, **especially** papers and books

- NP_{hyper} such as $\{NP_{hypo},\}^*\{(and|or)\}NP_{hypo}$
- such NP_{hyper} as $\{NP_{hypo},\}^*\{(and|or)\}NP_{hypo}$
- $NP_{hypo}\{, NP\}^*\{, \}$ or other NP_{hyper}
- $NP_{hypo}\{, NP\}^*\{, \}$ and other NP_{hyper}
- NP_{hyper} especially $\{NP_{hypo},\}^*\{(and|or)\}NP_{hypo}$



Taxonomy - Hierarchical Clustering

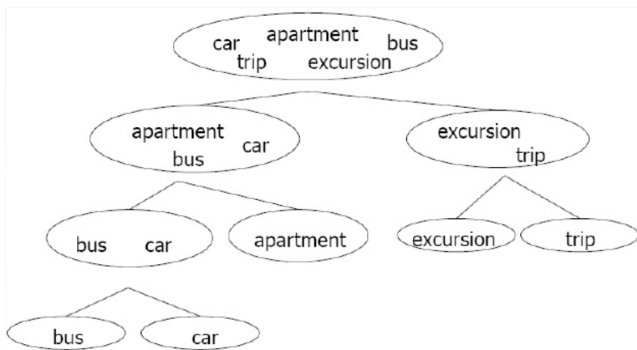
	Book _{obj}	Rent _{obj}	Drive _{obj}	Ride _{obj}	Join _{obj}
Hotel	x				
Apartment	x	x			
Car	x	x	x		
Bike	x	x	x	x	
Excursion	x				x
Trip	x				x

$$\text{Jaccard coefficient distance} = \frac{|A \setminus B|}{|A \cup B|}$$

	Hotel	Apartment	Car	Bike	Excursion	Trip
Hotel	1.0	0.5	0.33	0.25	0.5	0.5
Apartment		1.0	0.66	0.5	0.33	0.33
Car			1.0	0.75	0.25	0.25
Bike				1.0	0.2	0.2
Excursion					1.0	1.0
Trip						1.0



Taxonomy - Hierarchical Clustering



see [staab2010handbook]



Taxonomy - Document-based subsumption

Term t_1 subsumes term t_2 [is-a(t_2, t_1)] if t_1 appears in all the documents in which t_2 appears

$$P(x|y) = \frac{n(x, y)}{n(y)}$$

Term x subsumes term y iff $P(x|y) = 1$, where
 $n(x, y) \rightarrow$ the number of documents in which x and y co-occur
 $n(y) \rightarrow$ the number of documents that contain y



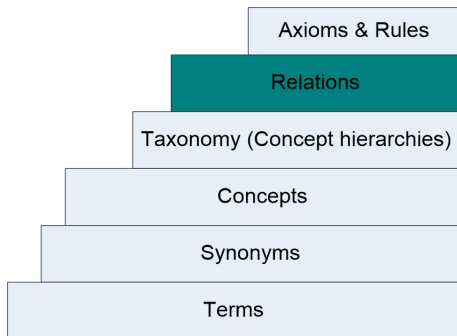
Learning relations

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Relation extraction

cure (domain:Doctor, range:Disease)



Relation extraction - Specific Relations

Discover anonymous associations between words

- X consists of Y (part-of)
The framework for OL consists of information extraction, ontology discovery and ontology organization
- X is used for Y (purpose)
OL is used for OE
- X leads to Y (causation)
Good OL methods lead to good OE



Relation extraction

OntoLT

Syntactic analysis: Maps a *subject* to the **domain**, the *predicate* or *verb* to **relation** and the *object* to its **range**.

The player kicked the ball to the net
relation: kick (domain: player, range: ball)

TextToOnto

$$\textit{love}(\textit{man}; \textit{woman}) \wedge \textit{love}(\textit{kid}; \textit{mother}) \wedge \textit{love}(\textit{kid}; \textit{grandfather})$$
$$\Rightarrow$$
$$\textit{love}(\textit{person}; \textit{person})$$

However, different verbs can represent *the same* (or a *similarTo*) relation

Clustering \rightarrow advise, teach, instruct



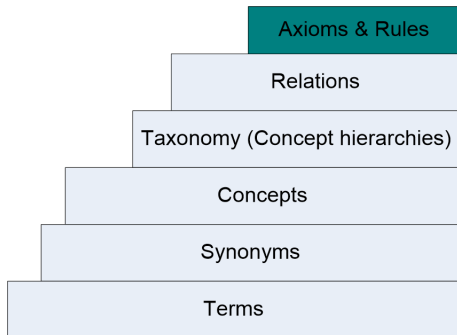
Learning rules and axioms

- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning**
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion



Rules and axioms extraction

$\forall x, y(\textit{sufferFrom}(x, y) \rightarrow \textit{ill}(x))$



Rule Extraction

DIRT - Discovery of Inference Rules from Text (Lin and Pantel, 2001)

- Let X be an algorithm which solves a problem Y
- Using similar constructions like **X solves Y** , **Y is solved by X** , **X resolves Y**
- $\forall x, y (solves(x, y) \Rightarrow isSolvedBy(y, x))$ (Inverse object property)
- $\forall x, y (solves(x, y) \Rightarrow resolves(x, y))$ (Equivalent object property)



Axiom Extraction

- Automated Evaluation of Ontologies - AEON (Völker et al., 2008)

Axioms are extracted (using lexico-syntactic patterns) from a Web Corpus

- Dealing with uncertainty and inconsistency (Haase and Völker, 2005)

Disjointness axioms \rightarrow disjoint(man,woman)



		Terms	Concepts	Taxonomic relations	Non-taxonomic relations
statistic methods	Text pre-processing	X			
	POS tagging	X			
	Sentence parsing	X			
	Latent semantic		X		
	Cooccurrence	X	X		
	Clustering		X	X	
	Term subsumption				X
Linguistic methods	Association rules				
	Seed words	X			
	Semantic lexicon		X	X	X
	Sub-categorization frames	X	X		
	Syntactic structure	X			X
	Dependency analysis	X			X
	Semantic templates			X	X
Logical methods	Lexico-syntactic patterns			X	X
	Axiom templates				
	Logical inference			X	X
	Inductive Logic				

Table: Ontology learning tasks and subtasks and the state-of-art techniques applied for each



- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation**
- 4 Ontology Learning Tools
- 5 Conclusion

Ontology Evaluation



Quality criteria

- **Accuracy** Does the ontology accurately model the domain?
- **Adaptability** Can the ontology easily be adapted to various uses?
- **Clarity** Is the meaning implied by the ontology clear?
- **Completeness** Does the ontology richly or thoroughly cover the domain?
- **Computational efficiency** How easily can automatic reasoners perform typical tasks?
- **Conciseness** Does the ontology include unnecessary axioms or assumptions?
- **Consistency** Does the ontology lead to logical errors or contradictions?
- **Organisational fitness** Is the ontology easily deployed in the application context? Is it easy to access? Does it align to other ontologies already in use?



How to evaluate OL

- Benchmark corpora and ontologies
- Evaluation of methods using different information sources



- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools**
- 5 Conclusion

Ontology Learning Tools



The *festival* **attracts** *culture* vultures to see live drama, dance and music

OntoLT

- *festival* and *culture* are class candidates - using statistical analysis (TF-IDF)
- **attracts** is a relation between festival and culture - using NLP



ASIUM - Acquisition of Semantic knowledge Using ML Methods

- Taxonomic relations among terms in technical texts
- Conceptual Clustering

OntoLearn

- Enrich a domain ontology with concepts and relations
- NLP and ML

Text-To-Onto

- Find taxonomic and non-taxonomic relations
- Statistics, Pruning Techniques and Association Rules
- Successor: Text2Onto tool



Text2Onto

- Ontology learning from textual documents **framework**
- System calculates a **confidence** for each learned object for better user interaction
- **Updates** the learned knowledge each time the corpus is changed and avoid processing it by scratch
- Interaction with end-users which is the central part of the architecture.
- Allows for easy
 - ① combination of algorithms,
 - ② execution of algorithms,
 - ③ writing new algorithms



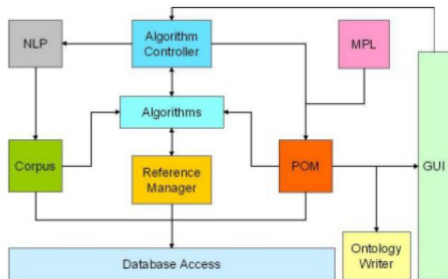
Text2Onto requirements

- Java 6 +
- WordNet
- GATE (General Architecture for Text Engineering)



Text2Onto components

- NLP engine
- Algorithms
- Algorithm Controller
- (Probabilistic Ontology Model) POM



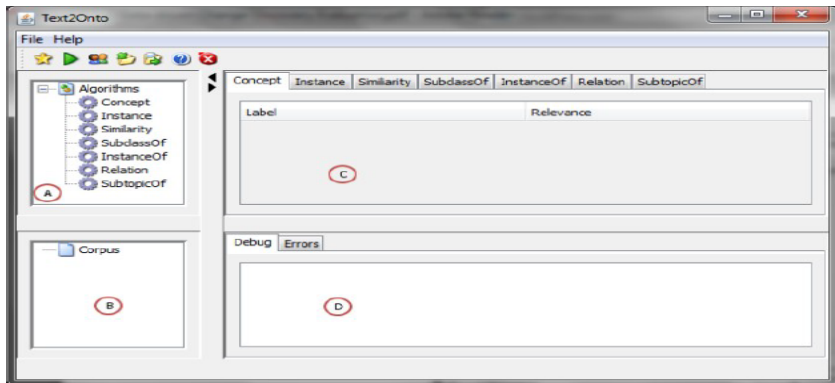
POM

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.

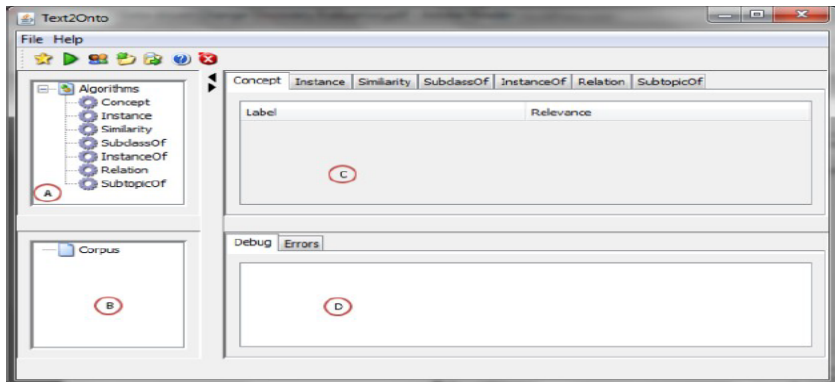


Text2Onto workflow



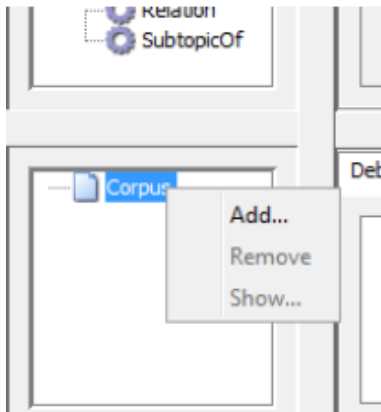
- (A) **Controller view** where we specify which Algorithms to use and how to combine the results of these algorithms.
- (B) **Corpus view** from where adding / removing a corpus is done.

Text2Onto workflow



- **(C) POM view panel.** Displays the results of the current ontology learning process.
- **(D) Displays** debugging messages and error messages.

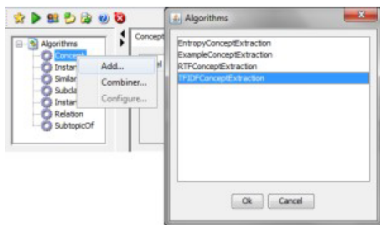
Text2Onto workflow



Step 1 - Add a Corpus

Right-click on the label *Corpus* on corpus view panel and add a corpus.

Text2Onto workflow



Step 2 - Specify algorithms to be applied

Right-click on the required entity on the controller view panel 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

Text2Onto workflow

The screenshot shows the Text2Onto application window. The 'Concepts' view panel is active, displaying a table of extracted concepts. The table has four columns: Domain, Range, and Confidence. The 'Internet' concept is highlighted in yellow.

Domain	Range	Confidence
fusion process	process	1.0
paper extract	product	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	0.5
member	part	0.5
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

Below the table, the 'Debug Errors' panel shows a list of ontology terms and a complex algorithm definition:

```

action, 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
elling world, knowledge management process, interface engine, modeling approach, student, staff, health insurance
company, process modeling, configure, category, uniform, process, ipbus, suit, note, group filesystem, label, st
ructure, online, interaction, solution, browsing, personal, integration, idea, paper extract, datum source, web
pr, class, agreement, format, world view, fusion process, creator, diary entry, access structure, categorization
, categorization scheme, mail, designer], class org.ontaware.text2onto.pom.POMInstanceOfRelation([instance-of s
emantic web, extension ], instance-of [semantic web, layer ], instance-of [word, product ], instance-of [business
engineering, modeling world ], instance-of [ methods, tool ]])

ComplexAlgorithm: SimilarityExtraction|combiner-org.ontaware.text2onto.algorithm.combiner.AverageCombiner.algor
ithm*[ContextSimilarityExtraction]
  
```

The results will appear on the POM view panel (C).

Text2Onto workflow

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).

Export the results

Results can be exported in KAON, RDFS or OWL format. To do this, go to File and click Export.

Text2Onto

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

Can Text2Onto help a user to build an ontology?



FRED

- Transforms natural language text into RDF/OWL with linked data support
- Based on ontology design patterns
- Graph visualization
- Export the RDF result: RDF/XML, Turtle, N3, etc.
- Tasks:
 - Entity linking to Semantic Web data
 - Relation extraction between frames, events, concepts and entities
 - Negation representation
 - Temporal relation extraction
 - ... more



- 1 Ontology and Ontology Learning
- 2 Methods of Ontology Learning
- 3 Ontology Evaluation
- 4 Ontology Learning Tools
- 5 Conclusion**

Conclusion



- We need to build Ontology quickly, easily and they have to be reliable!
- Fully automated OL system that works perfectly, doesn't exist **YET**
- User revision and interaction is essential
- Methods are based mainly on NLP techniques complemented with statistical measures

Ontology Learning is the old new era of developing ontologies. It is linked with many CS fields and it is all about understanding the reality through the structure of things

