

Knowledge support for Modeling and Simulation intermediate results of a dissertation project

Michal Šev enko,
Czech Technical University in Prague
Dept. of Computer Science

Knowledge Engineering Group Presentation,
Prague, March 22nd, 2007

the
KSMSA
project

Outline

- 1. Motivation, project goals, relation to the Semantic Web**
- 2. Basic components of the system**
 - a. System Architecture**
 - b. Documentation Framework**
 - c. Ontology Framework**
- 3. Experiments**
 - a. Automatic Classification of Documents**
 - b. Higher-order Inference**

Motivation – State of the Art

Current situation of work with scientific documentation

- Many tools for authoring and management of scientific documents, often need for combining several tools.
- Presentation of documents on WWW using purely presentation formats like HTML, PDF, MS Word.
- Global search engines see only these presentation-oriented documents → inaccurate search.

Motivation – Current Development

Developing areas:

- **XML-based formats: Semantic Web (RDF, OWL, ...), document representation (DocBook, MathML, SVG, ...)**
- **Development of ontologies for various domains.**
- **New generation of WWW (Semantic web) promises significant improvement of services, esp. search services.**

Motivation – Bottlenecks of the Semantic Web

- Standards are very complicated, low-level, lack of integrating applications.
- Decentralistic approach: “Let's not develop any upper, centrally accepted ontologies”.
- So far tool for researchers rather than for general public.

Motivation – Project Goals

Project KSMSA: Knowledge Support for Modeling and Simulation

The goal of the project is

- to create methodology for management of knowledge base in the form of large collection of scientific documents
- validate the methodology using prototype implementation of knowledge-based system

Methodology

The methodology is based on the idea of integration of the following processes:

- authoring documents
- connecting documents to the knowledge base (annotation)
- searching the knowledge base
- evolution of the semantic model of the knowledge

KSMSA System Architecture



Knowledge System – Ontology

The heart of the knowledge model is *ontology* – formal description of concepts in certain domain of interest. There are different degrees of formalization of ontology concepts:

1. Concepts provide fixed points for associating further structures (sense disambiguation).
2. Organization into taxonomies: navigation aid, simple inference (generalization & specialization).
3. More complex formalizations – more sophisticated inference such as problem solving – problematic.

Our knowledge model should support all degrees of formalization.

What is Ontology?

*large,
informal
ontologies*

WordNet, ...

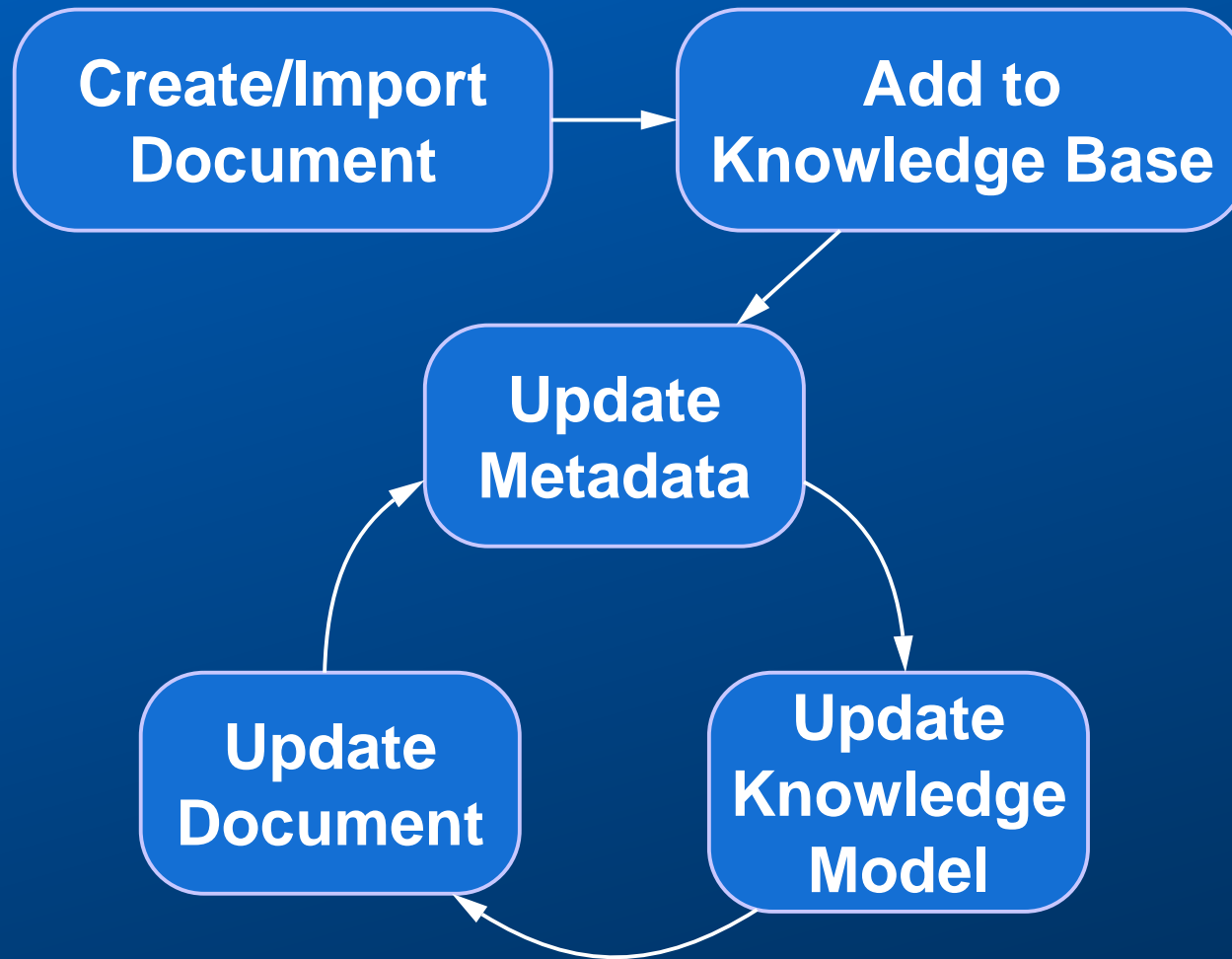
*higher-order,
richly-formalized
ontologies*

SUMO, CyCL, ...

*first-order,
frame-based
ontologies*

RDF, OWL, ...

Document Life Cycle



Software Tools

Tools needed for realization of our knowledge system:

- **Document management tools**
- **New knowledge modeling language**
- **Ontology authoring and browsing tools**
- **Knowledge base management tools**
- **Search of the knowledge base**

⇒ need for integration of these tools

Tools – Documentation Framework: RichDoc

- Support for basic documentation structures (sections, lists, tables)
- Support for advanced structures (figures, animations, mathematical typesetting)
- Independence of a document on its presentation (like HTML–CSS)
- Compact, extensible XML-based representation
- Scalability – supports large documents
- Possibility of embedding small documentation fragments into superordinate units
- Possibility of custom data integration, (metadata, ontology-based concepts)

Tools – Documentation Framework

Book Editor

File View Edit Insert Document Tools Help

History

Title	Modified
Modeling of Multidisciplinary 2 minutes	

- Modeling of Multidisciplinary Systems
 - About this course
 - Module 1 Mechanical rectilinear systems
 - Module 2 Rotational systems
 - Module 3 Coupled mechanical systems
 - Module 4 Electrical systems**
 - Module 5 Fluid systems
 - Module 6 Thermal systems

π ab

- DisplayedMath
 - π or
- DisplayedMath
 - π where $v(t_0)$ is the voltage across the capacitance
 - π For example, it may be recalled from physics that
- DisplayedMath
 - π where C [F] is the capacitance, ϵ [F/m] is the permittivity
 - π The capacitance of various real structures depends on geometry
 - π When charge is caused to flow into a capacitance
- DisplayedMath
 - π The process of energy storage is reversible, and
- DisplayedMath

current $i = dq/dt$, and the defining equation for a pure capacitor (4.11), the energy stored in an ideal capacitance is obtained from (2-79) with $q = Cv$:

$$\mathcal{E}_e = C \int_0^v v \cdot dv = \frac{1}{2} Cv^2 = \frac{1}{2} \frac{q^2}{C}$$

The process of energy storage is reversible, and all the electrical energy stored in an ideal capacitor is retrievable. Note that the energy stored depends neither on the sign of the voltage nor on the instantaneous value of the current:

$$\mathcal{E}_e \geq 0.$$

4.5 Pure electrical inductor

When current flows through a conductor, a magnetic field is established in the space or the material around the conductor. If this current is changed as a function of time, the intensity of the magnetic field will also vary with time. According to Lenz's law, this changing field will induce voltage differences in the conductor which will tend to oppose the changing current.

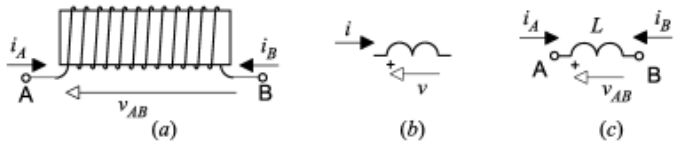


Figure 4.5 (a) Electrical coil. (b) Pure electrical inductor. (c) Coil model.

An electrical element is said to be a pure electrical inductor provided that the

Knowledge Modeling Language

SUO-KIF

- + **Expressive Power**
- + **Legible Syntax**

OWL

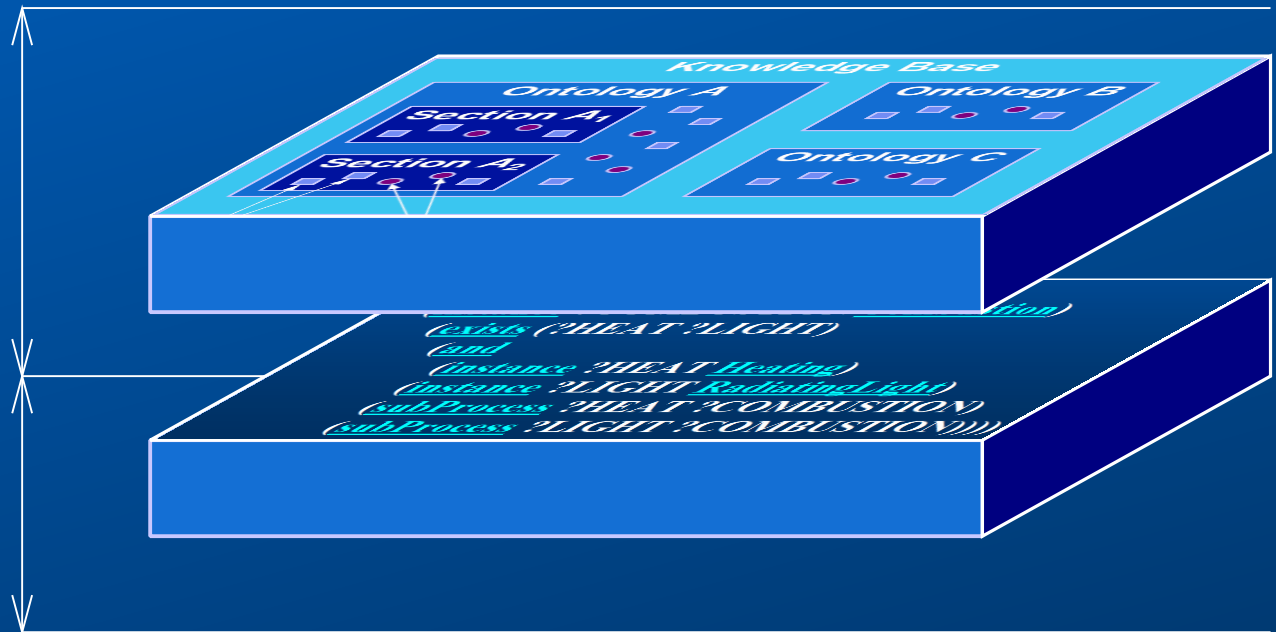
- + **Namespace Support**
- + **Modularity**

**New
Knowledge
Modeling
Language**

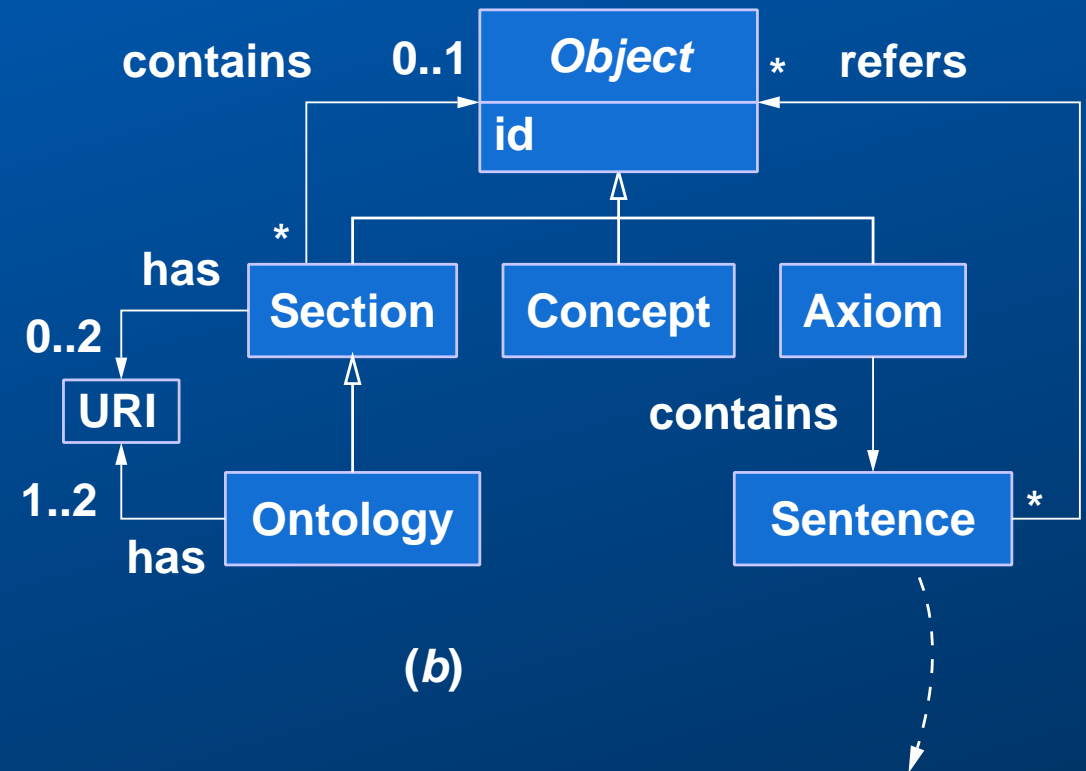
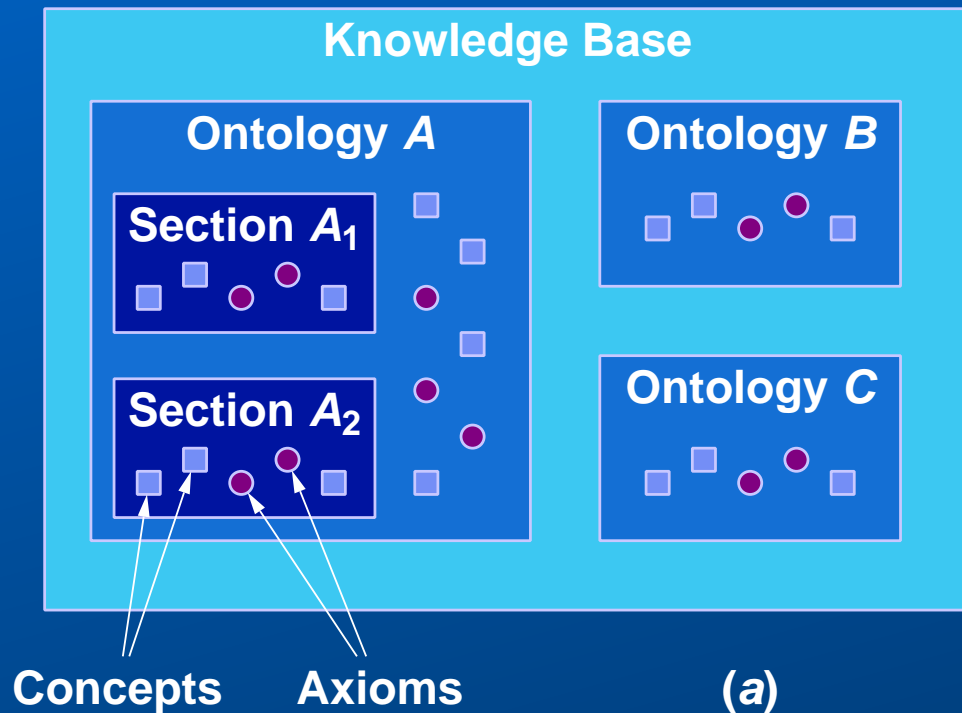
Knowledge Modeling Language

meta-level
(ontologies, sections,
namespaces)
SUO-KIF extension

axiomatic level
(first-order sentences)
equivalent to SUO-KIF



Knowledge Modeling Language – Meta-level



SUO-KIF sentences

```

=>
(instance ?COMBUSTION Combustion)
(exists (?HEAT ?LIGHT)
  (and
    (instance ?HEAT Heating)
    (instance ?LIGHT RadiatingLight)
    (subProcess ?HEAT ?COMBUSTION)
    (subProcess ?LIGHT ?COMBUSTION))))
  
```

Tools – Ontology Management

- Support for management of large ontologies
- Uniform representation of simple axioms (facts) as well as complex logical formulas in first-order or higher-order logic
- Special support for authoring particular axioms (subclass or instance relations, etc)
- Compact XML-based representation
- Extensible plugin architecture – advanced users may provide GUI support for certain types of axioms
- Supports importing (SUMO, WordNet) and exporting (KIF, OWL) of various formats.

Tools – Ontology Management

The screenshot displays the 'Ontology Browser: Concept Human' window. The left pane shows a tree view of the ontology hierarchy, with 'Human' selected under 'Hominid'. The right pane shows the hierarchical structure of 'Human' as a concept, with a description: 'Modern man, the only remaining species of the Homo genus.' The hierarchy is as follows:

- Entity
- Physical
- Object
- SelfConnectedObject
- CorpuscularObject
- OrganicObject
- Agent
- Organism
- Animal
- Vertebrate
- WarmBloodedVertebrate
- Mammal
- Primate
- Hominid
- Human

Additionally, 'Human' is shown as a sub-ordinate of 'SentientAgent' and 'CognitiveAgent'.

```
graph TD; Entity --> Physical; Physical --> Object; Object --> SelfConnectedObject; Object --> Agent; SelfConnectedObject --> CorpuscularObject; CorpuscularObject --> OrganicObject; OrganicObject --> Organism; Organism --> Animal; Animal --> Vertebrate; Vertebrate --> WarmBloodedVertebrate; WarmBloodedVertebrate --> Mammal; Mammal --> Primate; Primate --> Hominid; Hominid --> Human; Agent --> SentientAgent; Agent --> CognitiveAgent; SentientAgent --> Human; CognitiveAgent --> Human;
```

Experiments

- **Automatic Document Classification**
classify documents into thematic topics according to their full text
- **Higher-order semantics and inference**
construct proofs for higher-order theories

Automatic Document Classification

- Uses large database of English words, classified into topics
- Database was created using the full text of the Wikipedia encyclopedia

Automatic Document Classification

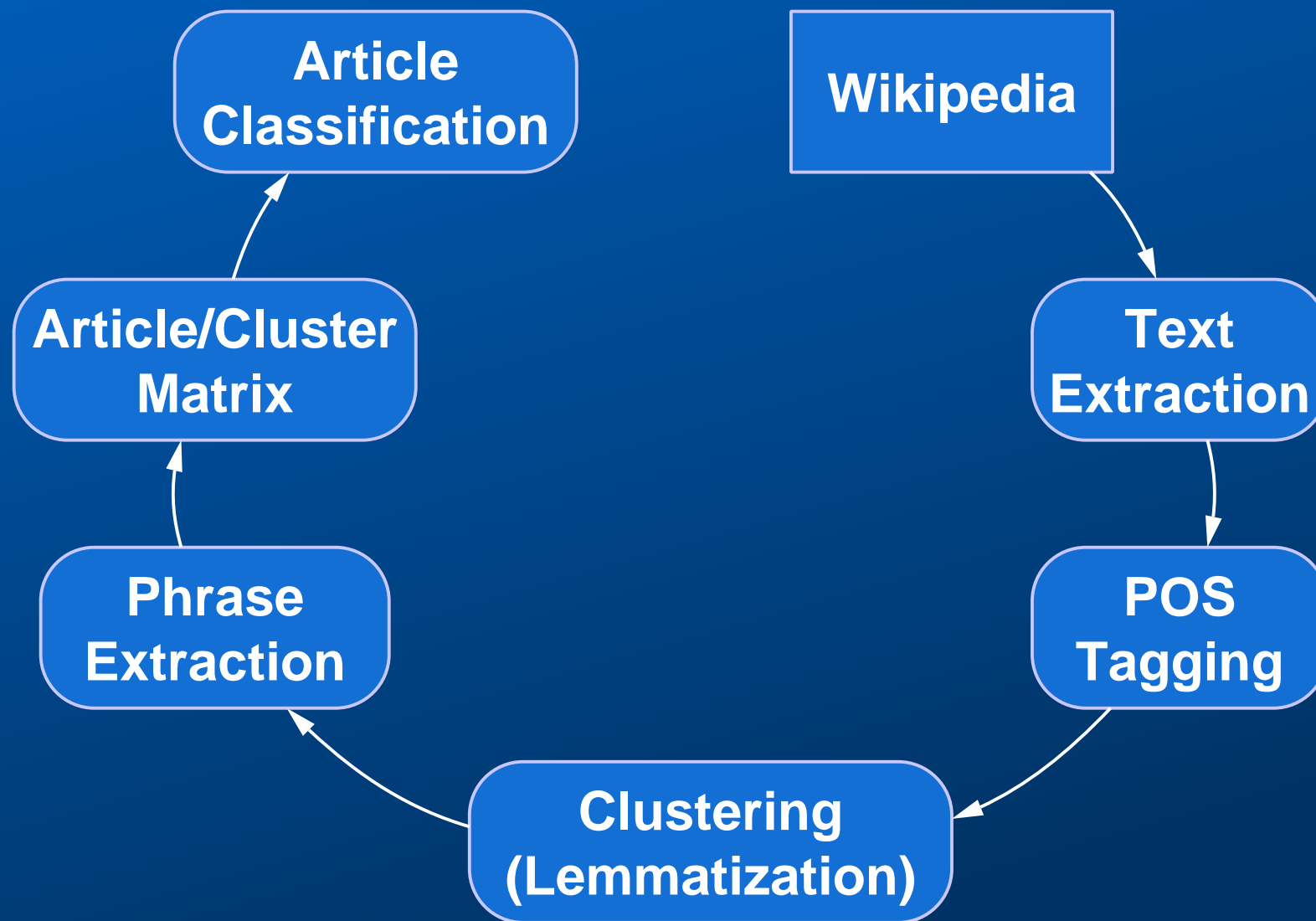
What information the corpus analysis provide?

- List of concepts that do not contribute to topic classification (“Semantic Stop List” – common words like *read*, *accept*, *computer* etc.
- For each category, list of concepts with high entropy gain.
- For each concept, list of categories with high entropy gain.

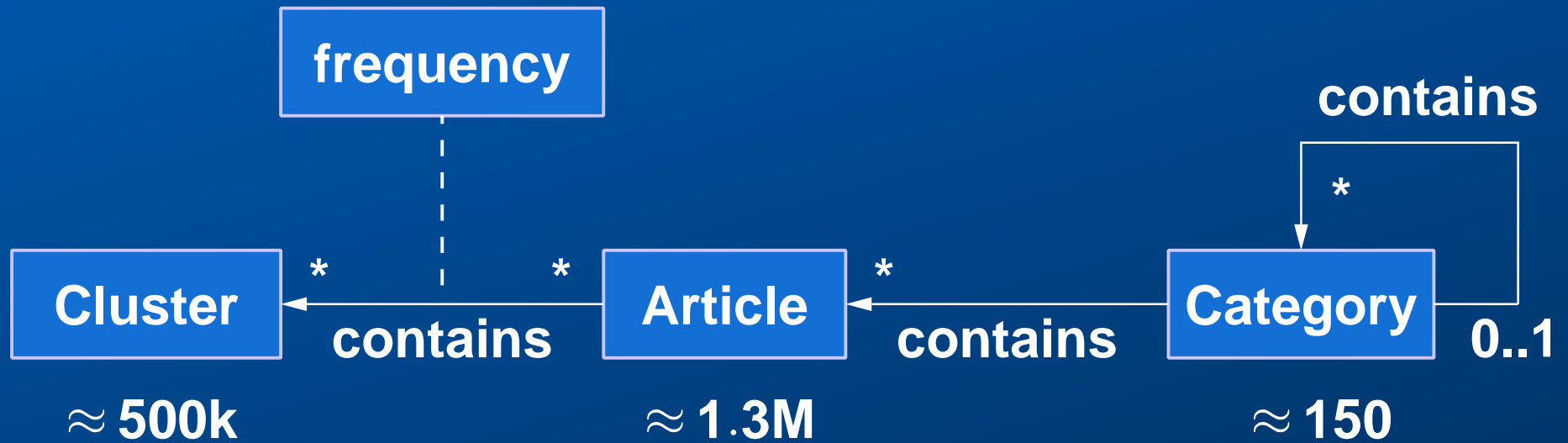
Why Wikipedia?

- Easily accessible large body of English text
- Relatively well-weighted content
- Relatively good classification into topical categories; *but*: category system is too dense and inaccurate, must be “distilled” into much smaller number of categories

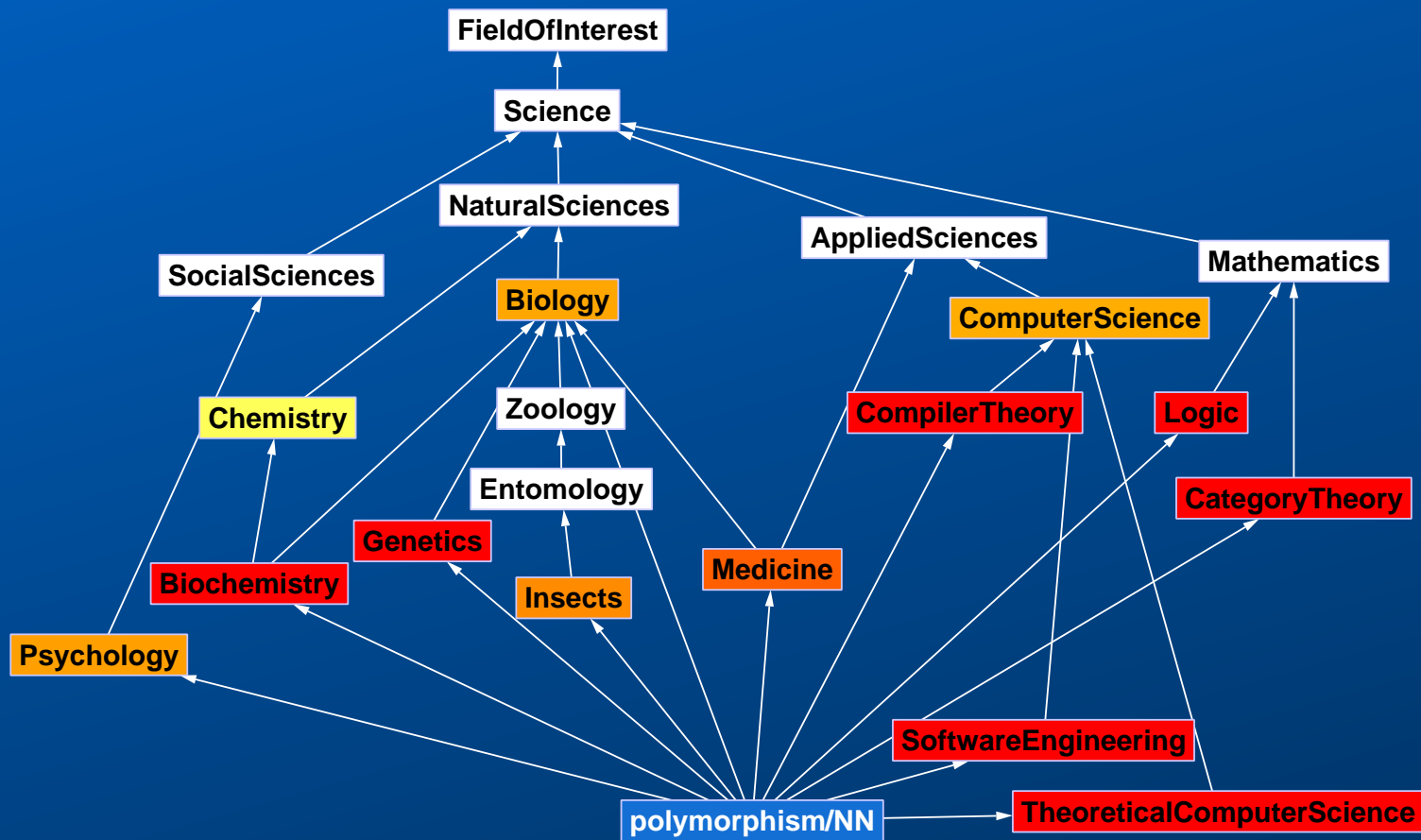
Wikipedia Corpus Analysis



Wikipedia Corpus Database



Wikipedia Corpus – Classification Example



Cluster “Polymorphism”

Significant topics: 88 of 1.34M
Entropy: 13.90

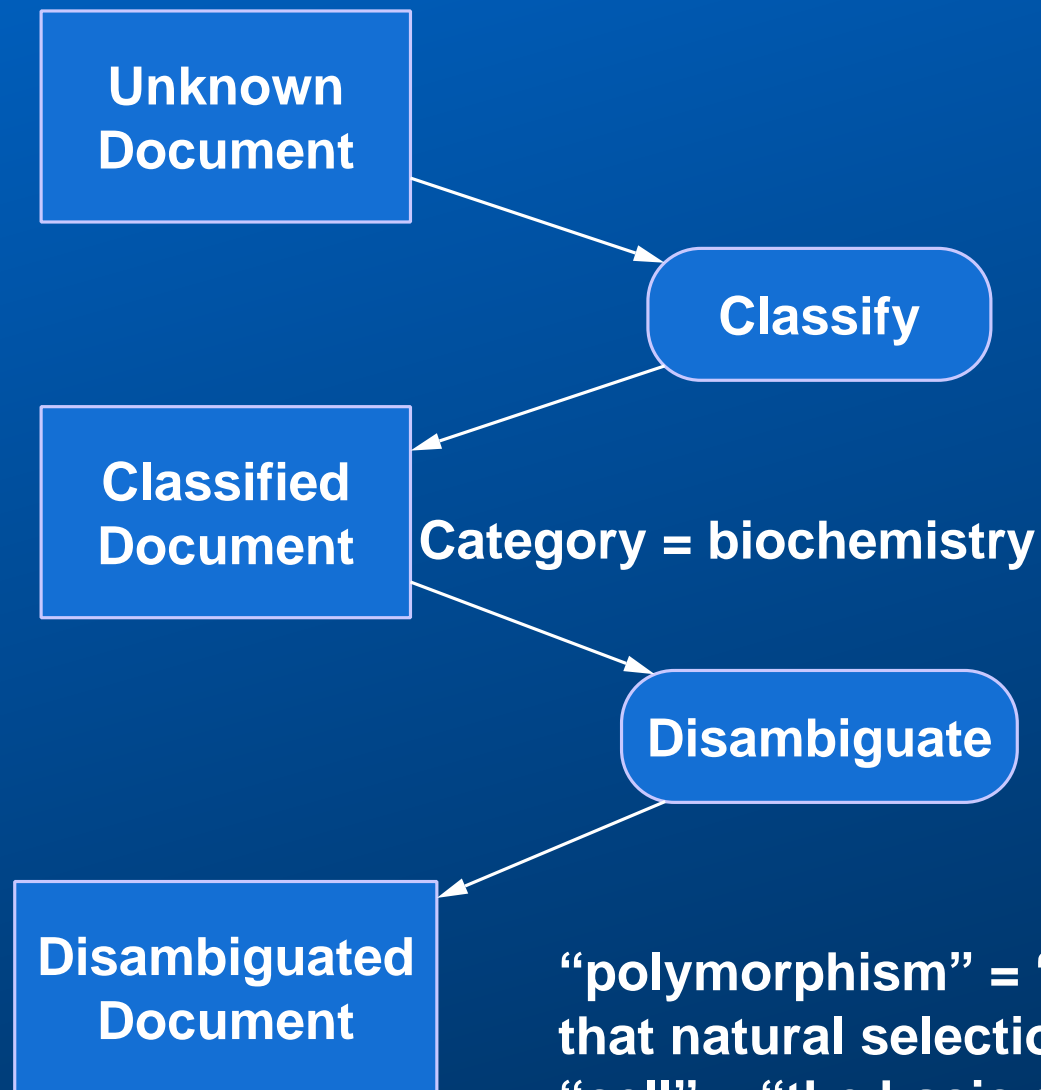
Category “Biochemistry”

Significant topics: 7 of 2.26k
Entropy: 8.34

Topics

- 5 Restriction fragment length polymorphism
- 3 Amplified fragment length polymorphism
- 3 Single nucleotide polymorphism
- 2 Polymerase chain reaction
- 2 Biochip
- 2 Perilipin
- 2 RAGE

Wikipedia Corpus – Topic-based disambiguation



“polymorphism” = “the genetic variation within a population that natural selection can operate on”

“cell” = “the basic structural and functional unit of all organisms” etc.

Wikipedia Corpus – Problems

- Topic classification sometimes does not work. Example: Wikipedia's category “Submarines” is mapped to category “Sports” according to shortest category path Submarines → Ships by type → Sailing → Olympic sports → Sports. More appropriate would be e.g. “Security”.
- The frequency-based phrase extraction extracts frequently-used collocations that are not concepts. For instance, the phrase “said to have been” appears in about 3000 articles, and is therefore identified as concept.

Higher-order Semantics

- The SUO-KIF language supports higher-order sentences, such as `believes(Mary, teacher(John))`, or `holdsDuring(Monday, located(John, Prague))`,
but does not provide any semantics or inference support for them
- We propose an extension of common first-order semantics, and an inference system that can construct proofs in higher-order theories

Higher-order Semantics

- Higher order atom: $P(\dots, \varphi, \dots)$, P is predicate, φ is sentence
- First order interpretation $\mathcal{M}: \Lambda \rightarrow \{\text{true}, \text{false}\}$, Λ is set of well-formed formulas; \mathcal{M} satisfies first-order consistency requirements, such as $\mathcal{M}(\varphi) \neq \mathcal{M}(\neg\varphi)$, etc.
- Higher order interpretation \mathcal{H} must fulfill those requirements of \mathcal{M} and
 1. if $\mathcal{H} \models P(\dots, \varphi, \dots)$, then $\mathcal{H} \not\models P(\dots, \neg\varphi, \dots)$
 2. if $\mathcal{H} \models P(\dots, \varphi, \dots)$, and $\varphi \models \psi$, then $\mathcal{H} \models P(\dots, \psi, \dots)$

Higher-order Resolution

This yields, besides standard resolution rule:

- $\{\varphi \vee A, \psi \vee \neg B\} \vdash (\varphi \vee \psi)\theta$, A and B are atoms, and θ is the most general unifier of A and B .

two additional resolution rules:

1. $\{\varphi \vee P(\dots, \alpha, \dots), \psi \vee \neg P(\dots, \beta, \dots)\} \vdash (\varphi \vee \psi)\theta$, if $\alpha\theta \vdash \beta\theta$; α and β are sentences, $\alpha\theta$ and $\beta\theta$ must be closed sentences (not containing free variables)
2. $\varphi \vee P(\dots, \psi, \dots) \vdash \varphi \vee \neg P(\dots, \neg \psi, \dots)$

Higher-order Resolution: Example 1

(1) Mary is in Prague during Tuesday, (2) Prague does not overlap with Berlin, (3) Tuesday Afternoon is part of Tuesday, (4) If someone is in I_1 and I_1 does not overlap with I_2 , then someone is not in I_2 , and (5) If something is true during t and t' is part of t , then something is true during t' . We want to prove that (q) Mary is not in Berlin during Tuesday Afternoon.

(1) $\text{holdsDuring}(\text{Tuesday}, \text{located}(\text{Mary}, \text{Prague}))$

(2) $\neg \text{overlaps}(\text{Prague}, \text{Berlin})$

(3) $\text{partOf}(\text{TuesdayAfternoon}, \text{Tuesday})$

(4) $\text{located}(x, I_1) \wedge \text{located}(x, I_2) \Rightarrow \text{overlaps}(I_1, I_2)$

(5) $\text{holdsDuring}(t, p) \wedge \text{partOf}(t', t) \Rightarrow \text{holdsDuring}(t', p)$

($\neg q$) $\text{holdsDuring}(\text{TuesdayAfternoon}, \text{located}(\text{Mary}, \text{Berlin}))$

Proof:

(t_1) = (3) \times (5) $\text{holdsDuring}(\text{Tuesday}, p) \Rightarrow \text{holdsDuring}(\text{TuesdayAfternoon}, p)$

(t_2) = (t_1) \times (1) $\text{holdsDuring}(\text{TuesdayAfternoon}, \text{located}(\text{Mary}, \text{Prague}))$

(t_3) = (t_2)^{*} $\neg \text{holdsDuring}(\text{TuesdayAfternoon}, \neg \text{located}(\text{Mary}, \text{Prague}))$

(t_4) = (t_3) \times ($\neg q$) \square if (a) $\text{located}(\text{Mary}, \text{Berlin}) \vdash (b) \neg \text{located}(\text{Mary}, \text{Prague})$

(t'_1) = (a) \times (4) $\text{located}(\text{Mary}, I_1) \Rightarrow \text{overlaps}(I_1, \text{Berlin})$

(t'_2) = (t'_1) \times (2) $\neg \text{located}(\text{Mary}, \text{Prague})$

(t'_3) = (t'_2) \times ($\neg b$) \square

Higher-order Resolution: Example 2

(1) Everyone knows whether he is a teacher or not. (2) Everyone believes what he knows. (3) What someone knows must be true. (4) Mary believes that John is a teacher. (5) John doubts he is a teacher. (q) There is a person that believes that some false proposition is true.

(1a) $\text{teacher}(x) \Rightarrow \text{knows}(x, \text{teacher}(x))$

(1b) $\neg \text{teacher}(x) \Rightarrow \text{knows}(x, \neg \text{teacher}(x))$

(2) $\text{knows}(x, p) \Rightarrow \text{believes}(x, p)$

(3) $\text{knows}(x, p) \Rightarrow p$

(4) $\text{believes}(\text{Mary}, \text{teacher}(\text{John}))$

(5) $\text{believes}(\text{John}, \neg \text{teacher}(\text{John}))$

($\neg q$) $\text{believes}(x, p) \Rightarrow p$

Proof:

(t_1) = (5)* $\neg \text{believes}(\text{John}, \text{teacher}(\text{John}))$

(t_2) = (t_1) \times (2) $\neg \text{knows}(\text{John}, \text{teacher}(\text{John}))$

(t_3) = (t_2) \times (1a) $\neg \text{teacher}(\text{John})$

(t_4) = (4) \times ($\neg q$) $\text{teacher}(\text{John})$

(t_5) = (t_4) \times (t_3) \square

The End...

Project KSMSA

<http://virtual.cvut.cz/ksmsaWeb/main>

RichDoc Framework

<http://richdoc.sourceforge.net>