



Knowledge Discovery over the Deep Web, Semantic Web and XML

Aparna S. Varde, Fabian M. Suchanek,
Richi Nayak and Pierre Senellart

DASFAA 2009, Brisbane, Australia

Introduction

- The Web is a vast source of information
- Various developments in the Web
 - Deep Web
 - Semantic Web
 - XML Mining
 - Domain-Specific Markup Languages
- These enhance knowledge discovery

Agenda

- Section 1: Deep Web
 - Slides by Pierre Senellart
- Section 2: Semantic Web
 - Slides by Fabian M. Suchanek
- Section 3: XML Mining
 - Slides by Richi Nayak
- Section 4: Domain-Specific Markup Languages
 - Slides by Aparna Varde
- Summary and Conclusions

Section 1: Deep Web

Pierre Senellart

Department of Computer Science and Networking

Telecom Paristech

Paris, France

pierre@senellart.com

What is the Deep Web

Definition (Deep Web, Hidden Web)

All the content of the Web that is not directly accessible through **hyperlinks**. In particular: HTML forms, Web services.



Size estimate

- [Bri00] 500 times more content than o
- Dozens of thousands of databases.
- [HPWC07] ~ 400 000 deep Web databases.

Sources of the Deep Web

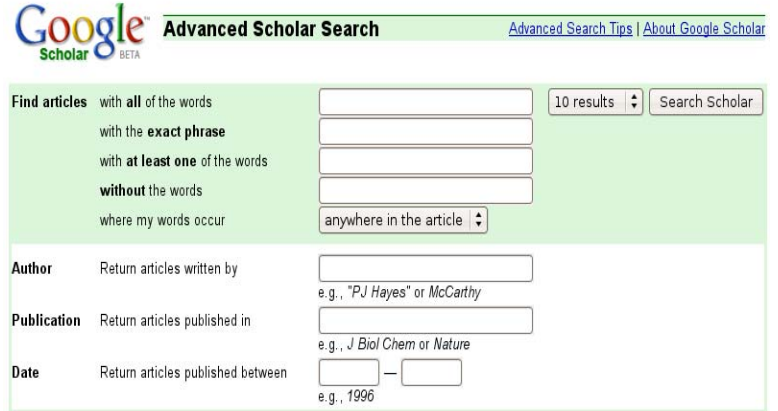
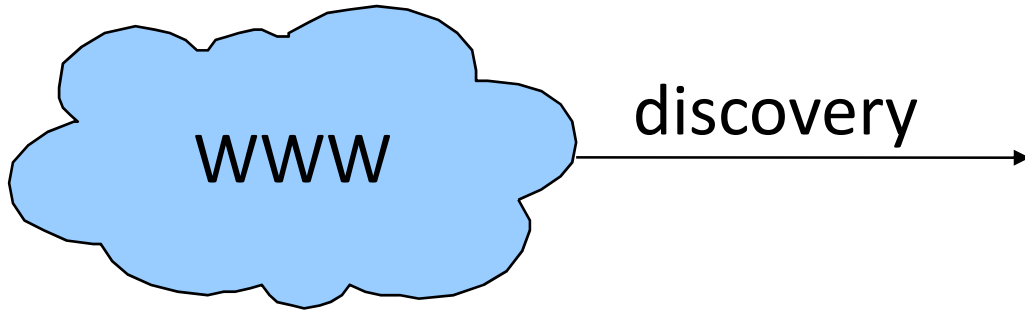
Examples

- *Yellow Pages* and other directories;
- Library catalogs;
- Publication databases;
- Weather services;
- Geolocalization services;
- US Census Bureau data;
- etc.

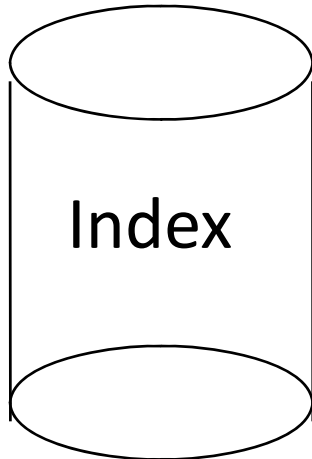
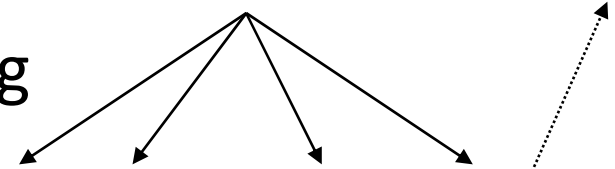
Discovering Knowledge from the Deep Web

- Content of the deep Web hidden to classical Web search engines (they just follow links)
- But very valuable and high quality!
- Even services allowing access through the surface Web (e.g., e-commerce) have more semantics when accessed from the deep Web
- How to **benefit** from this information?
- How to do it **automatically**, in an **unsupervised** way?

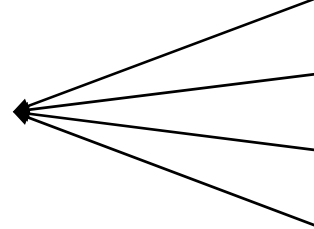
Extensional Approach



siphoning



indexing



Google Scholar BETA

Web Images Video News Maps more

data

Search

Advanced Scholar Search

Scholar Professor

Scholar Help

14 seconds

Scholar All articles - Recent articles

Results 1 - 10 of about 29,900 for monoid. (0.11 seconds)

On finite monoid

MP Schützenberger - ... 1965) On Finite I definition is given to Cited by 267 - Relat

System identification: th

Ljung - 1986 - Prentice-Hall, Inc. U Cited by 7815 - Related articles - Web

Nonlinear systems

HK Khalil, JW Grizzle - 1996 - div pr Cited by 261 - Related articles - Web

Finite monoids

DAM Barrington, D ... 2. Background a binary operation an Cited by 139 - Relat

Relative

J Almeida, P Weil - Cited by 68 - Relat

Rational

S Eilenberg, MP St ... imprints - New York - Wiley 330 Cited by 2003 - Related articles - Web

SCHÜTZENBERG

MacLane Received ... Cited by 131 - Relat

Neural networks and physical

JJ Hopfield - Proceedings of the nat ... -omeurs). The physical meaning of by an appropriate phase space flow Cited by 2723 - Related articles - Web

Word problems and a homological finiteness condition for monoids

Cited by 2723 - Related articles - Web

7 seconds

bootstrap

Notes on the Extensional Approach

- Main issues:
 - Discovering services
 - Choosing appropriate data to submit forms
 - Use of data found in result pages to bootstrap the siphoning process
 - Ensure good coverage of the database
- Approach **favored by Google** [MHC+06], used in production
- Not always feasible (huge load on Web servers)

Notes on the Extensional Approach

- Main issues:
 - Discovering services
 - Choosing appropriate data to submit forms
 - Use of data found in result pages to bootstrap the siphoning process
 - Ensure good coverage of the database
- Approach **favored by Google** [MHC+06], used in production
- Not always feasible (huge load on Web servers)

Intensional Approach



discovery



Google Scholar BETA Advanced Scholar Search [Advanced Search Tips](#) | [About Google Scholar](#)

Find articles with **all of the words** 10 results

with the **exact phrase**

with **at least one** of the words

without the words

where my words occur anywhere in the article

Author Return articles written by
e.g., "PJ Hayes" or McCarthy

Publication Return articles published in
e.g., J Biol Chem or Nature

Date Return articles published between -
e.g., 1996

probing



Google Scholar Web Images Video News Maps more »
data Search [Advanced Scholar Search](#)
[Submit Feedback](#)
[Scholar Help](#)

Scholar All articles - [Recent articles](#) Results 1 - 10 of about 91,400,000 for [data](#) [definition] (0.14 seconds)

1 Fisher R [The use of multiple measurements in taxonomic problems](#)
JE Psychol, ACO Generalis, SA Genet, M Biol, BMC ... - Ann of Eugenics, 1936 - biomedcentral.com
... Culhane A, Perniere G, Considine E, Cotter T, Higgins D. [Between-group analysis of microarray data](#) ... Comput Stat Data Anal 2004, 46:407-425 ...
[Cited by 3952](#) - [Related articles](#) - [Cached](#) - [Web Search](#)

[The protein kinase encoded by the Afl proto-oncogene is a target of the PDGF-activated...](#)
... Franke, Si Yang, To Chan, K Data, A Kazlauskas, Dh ... - Cell(Cambridge), 1995 - cat.inist.fr
TF FRANK, SUNG-IL YANG, TO CHAN, K DATA, A KAZLAUSKAS, DK MORRISON, DR KAPLAN, PN TSICHLIS Cell(Cambridge) 81:55, 727-736, Cell Press, 1995.
[Cited by 1409](#) - [Related articles](#) - [Web Search](#) - [BI Direct](#) - [All 4 versions](#)

[RAG-2-deficient mice lack mature lymphocytes owing to inability to initiate V\(D\)J rearrangement](#)
FB Pollock, DP Policy J Subscribers, J ... - Cell, 1992 - cell.com
... Both genetic and biochemical [data](#) point toward a physiological role for this complex as the elusive hairpin-opening activity in V(D)J recombination ...
[Cited by 1316](#) - [Related articles](#) - [Cached](#) - [Web Search](#) - [All 4 versions](#)

[Random data analysis and measurement procedures](#)
JS Bendat, AD Piersol - Measurement Science and Technology, 2000 - iop.org
BOOK REVIEW: [Random Data Analysis and Measurement Procedures](#) ... Chapter ten deals with [data](#) acquisition and processing, including [data](#) qualification ...
[Cited by 3941](#) - [Related articles](#) - [Web Search](#) - [SI/DOC Catalogue](#) - [All 10 versions](#)

[Data mining: practical machine learning tools and techniques with Java implementations](#) - [walkato.ac.nz](#) [pdf](#)
Witten, E Frank - ACM SIGMOD Record, 2002 - portal.acm.org
[Data Mining: Practical Machine Learning Tools and ...](#) Witten and Frank's textbook was

analyzing



Form wrapped as a Web service

query



Notes on the Intensional Approach

- More **ambitious** [CHZ05, SMM+08]
- Main issues:
 - Discovering services
 - Understanding the structure and semantics of a form
 - Understanding the structure and semantics of result pages (wrapper induction)
 - Semantic analysis of the service as a whole
- No significant load imposed on Web servers

Discovering deep Web forms

- Crawling the Web and selecting forms
- But **not all forms!**
 - Hotel reservation
 - Mailing list management
 - Search within a Web site
- **Heuristics:** prefer GET to POST, no password, no credit card number, more than one field, etc.
- Given domain of interest: use **focused crawling** to restrict to this domain

Web forms

Authors	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Title	<input type="text"/>		Year <input type="text"/>	Page <input type="text"/>
Conference	<input type="text"/>	ID <input type="text"/>		
Journal	<input type="text"/>	Volume <input type="text"/>	Number <input type="text"/>	
<input type="button" value="Search"/>	<input type="button" value="Reset"/>	Maximum of <input type="text" value="100"/> matches		

- **Simplest case:** associate each form field with some **domain concept**
- **Assumption:** fields independent from each other (not always true!), can be queried with words that are part of a **domain instance**

Structural analysis of a form (1/2)

- 1) Build a **context** for each field:
 - label tag;
 - id and name attributes;
 - text immediately before the field.
- 2) Remove **stop words, stem**
- 3) **Match** this context with concept names or concept ontology
- 4) Obtain in this way **candidate annotations**

Structural analysis of a form (1/2)

- 1) Build a **context** for each field:
 - label tag;
 - id and name attributes;
 - text immediately before the field.
- 2) Remove **stop words, stem**
- 3) **Match** this context with concept names or concept ontology
- 4) Obtain in this way **candidate annotations**

Structural analysis of a form (2/2)

For each field annotated with concept c :

- 1) Probe the field with nonsense word to get an **error page**
- 2) **Probe** the field with instances of concept c
- 3) Compare pages obtained by probing with the error page (e.g., clustering along the DOM tree structure of the pages), to distinguish error pages and **result pages**
- 4) **Confirm** the annotation if enough result pages are obtained

Structural analysis of a form (2/2)

For each field annotated with concept c :

- 1) Probe the field with nonsense word to get an **error page**
- 2) **Probe** the field with instances of concept c
- 3) Compare pages obtained by probing with the error page (e.g., clustering along the DOM tree structure of the pages), to distinguish error pages and **result pages**
- 4) **Confirm** the annotation if enough result pages are obtained

Bootstrapping the siphoning

- Siphoning (or probing) a deep Web database requires many relevant data to submit the form with
- **Idea:** use **most frequent words** in the content of the result pages
- Allows **bootstrapping** the siphoning with just a few words!

Inducing wrappers from result pages

Pages resulting from a given form submission:

- share the **same structure**
- set of **records** with fields
- **unknown** presentation!

Find: remi gilleron Documents Citations

Searching for PHRASE remi gilleron.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP

7 documents found. Order: number of citations.

[PAC Learning under Helpful Distributions - Denis, Gilleron \(1997\)](#) (Correct) (10 citations)

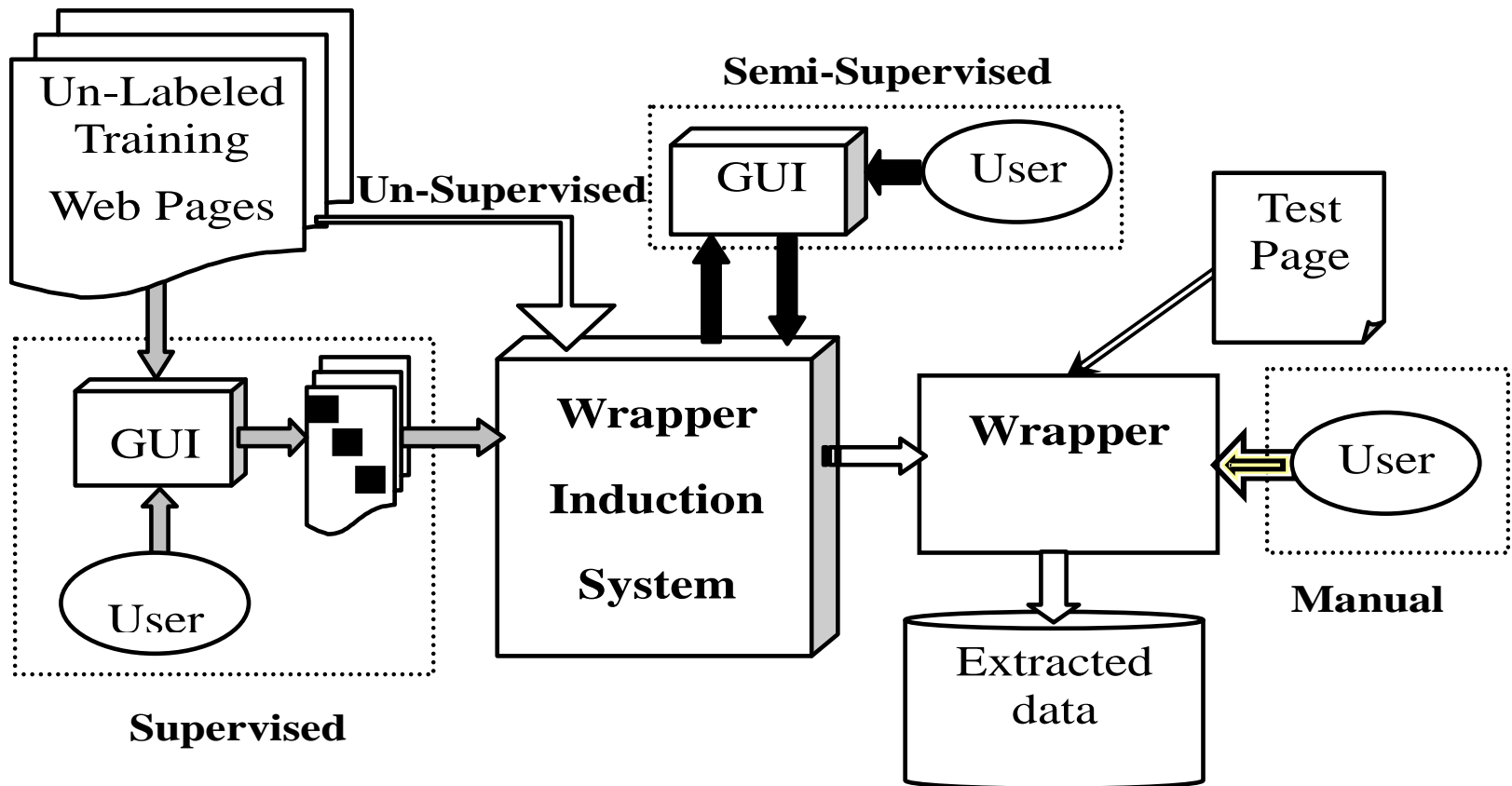
Helpful Distributions y Francois Denis, Remi Gilleron LIFL, URA 369 CNRS, Universit'e de Lille 1 59655 1 59655 Villeneuve d'Ascq FRANCE e-mail: denis.gilleron@lifl.fr Abstract A PAC model under helpful on Algorithmic Learning Theory ALT'97 (Denis and Gilleron, 1997) Introduction it seems that many

Sort by T-Meter	Sort by Title	Sort by Year
81%	Grindhouse Director Screenwriter Producer	2007
- N/A	Death Proof Director	2007
* 59%	Hostel Executive Producer	2006
- N/A	Reservoir Dogs/Bad Lieutenant Director	2006
- N/A	Inglorious Bastards Director	2006
97%	Double Dare Featured	2005
78%	Sin City Additional Directing	2005
* 29%	The Muppets: Wizard Of Oz Star	2005
* 0%	Daltry Calhoun Executive Producer	2005
85%	Kill Bill Vol. 2 Director Screenwriter	2004
100%	Z Channel: A Magnificent Obsession Featured	2004
85%	Kill Bill Vol. 1 Director Screenwriter Producer	2003

Goal

Building **wrappers** for a given kind of result pages, in a fully automatic way.

Information extraction systems [CKGS06]



Unsupervised Wrapper Induction

- Use the (repetitive) structure of the result pages to infer a **wrapper** for all pages of this type
- Possibly: use in parallel with **annotation** by recognized concept instances to learn with **both the structure and the content**



Some perspectives

- Dealing with **complex forms** (fields allowing Boolean operators, dependencies between fields, etc.)
- **Static analysis** of JavaScript code to determine which fields of a form are required, etc.
- A lot of this is also applicable to **Web 2.0/AJAX** applications

References

- [Bri00] BrightPlanet. **The deep Web: Surfacing hidden value**. White paper, July 2000.
- [CHZ05] K. C.-C. Chang, B. He, and Z. Zhang. **Towards large scale integration: Building a metaquerier over databases on the Web**. In *Proc. CIDR*, Asilomar, USA, Jan. 2005.
- [CKGS06] C.-H. Chang, M. Kaye, M. R. Girgis, and K. F. Shaalan. **A survey of Web information extraction systems**. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411-1428, Oct. 2006.
- [CMM01] V. Crescenzi, G. Mecca, and P. Merialdo. **Roadrunner: Towards automatic data extraction from large Web sites**. In *Proc. VLDB*, Roma, Italy, Sep. 2001.
- [HPWC07] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. **Accessing the deep Web: A survey**. *Communications of the ACM*, 50(2):94–101 May 2007.
- [MHC+06] J. Madhavan, A. Y. Halevy, S. Cohen, X. Dong, S. R. Jeffery, D. Ko, and C. Yu. **Structured data meets the Web: A few observations**. *IEEE Data Engineering Bulletin*, 29(4):19–26, Dec. 2006.
- [SMM+08] P. Senellart, A. Mittal, D. Muschick, R. Gilleron et M. Tommasi, **Automatic Wrapper Induction from Hidden-Web Sources with Domain Knowledge**. In *Proc. WIDM*, Napa, USA, Oct. 2008.

Section 2: Semantic Web

Fabian M. Suchanek

Databases and Information Systems

Max Planck Institute for Informatics

Saarbrücken, Germany

suchanek@mpi-inf.mpg.de

Motivation



scientists from Brisbane

[Australia's scientists visit Brisbane](#)

The National Science Education Unit invites Australian scientists to gather in Brisbane

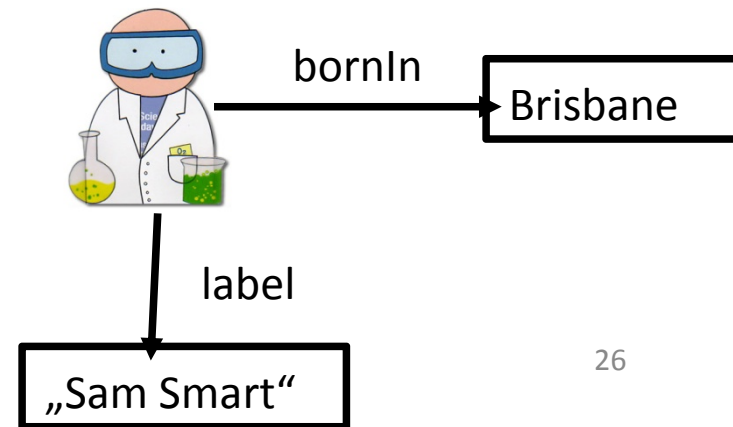
www.nsceu.au/brisbane

Today's state of the art

```
<HTML>  
  Sam Smart is a scientist from  
  Brisbane.  
</HTML>
```



Vision of the Sematic Web



The Semantic Web

The Semantic Web is the project of creating a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

Goals:

- make computers „understand“ the data they store
- allow them to answer „semantic“ queries
- allow them to share information across different systems

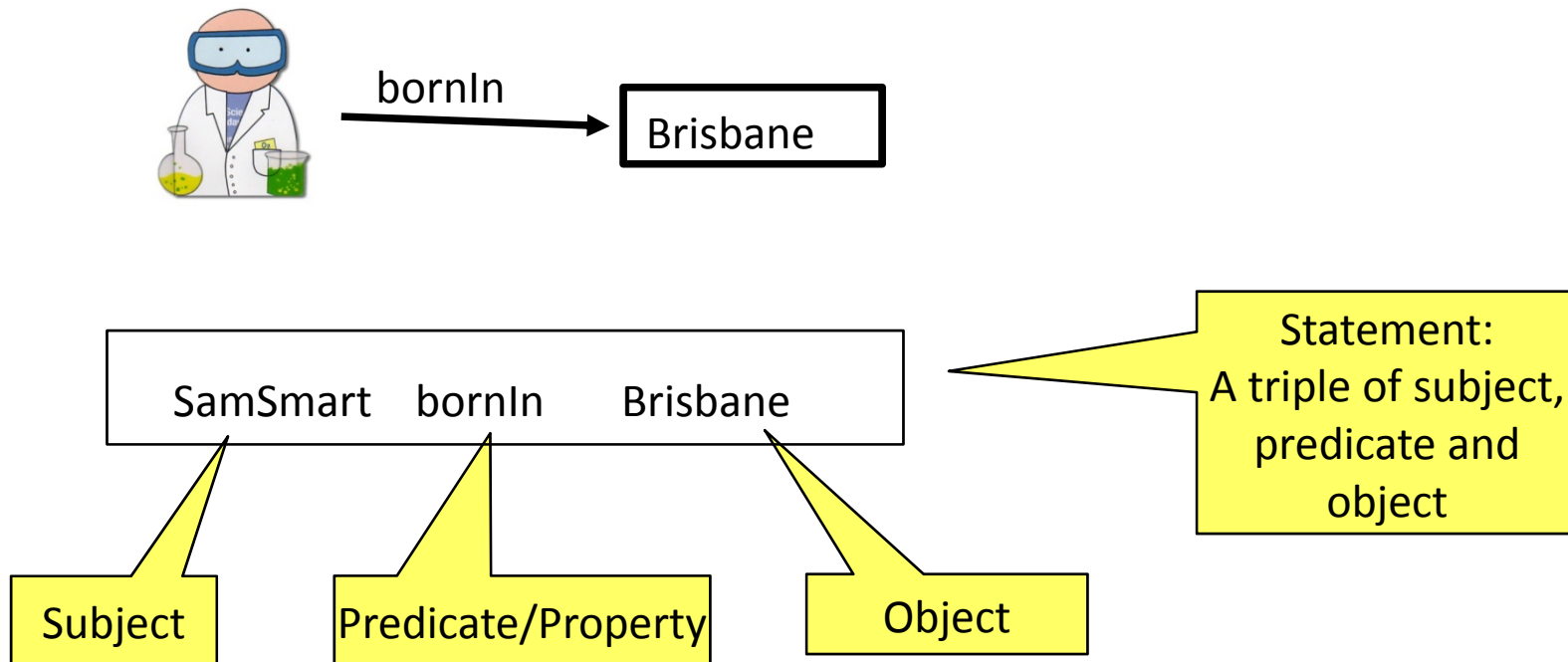
Techniques: (= this talk)

- defining semantics in a machine-readable way (RDFS)
- identifying entities in a globally unique way (URIs)
- defining logical consistency in a uniform way (OWL)
- linking together existing resources (LOD)

<http://www.w3.org/2001/sw/>

The Resource Description Framework (RDF)

RDF is a format of knowledge representation that is similar to the Entity-Relationship-Model.



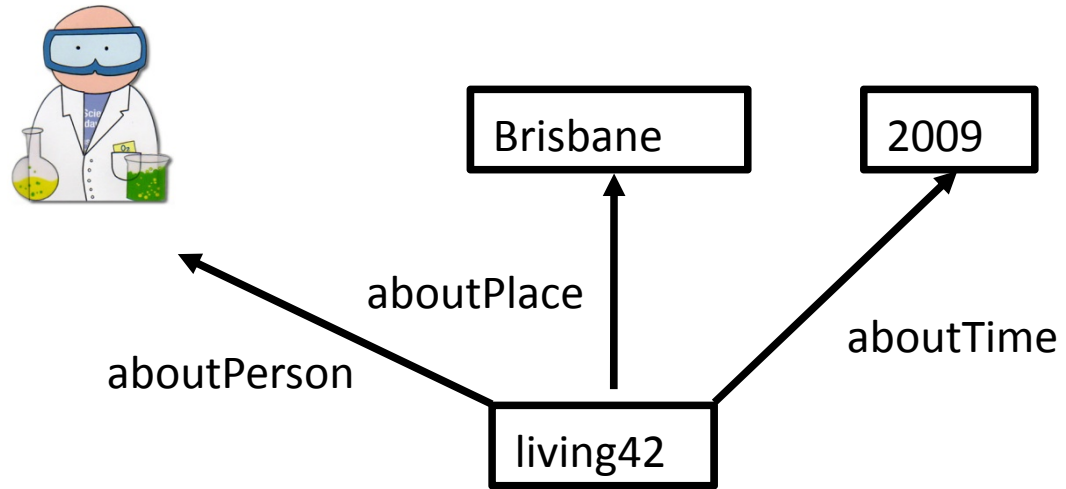
<http://www.w3.org/TR/rdf-prior/>

RDF is used as the only knowledge representation language.

=> All information is represented in a simple, homogeneous, computer-processable way.

n-ary relationships

n-ary relationships can always be reduced to binary relationships by introducing a new identifier.

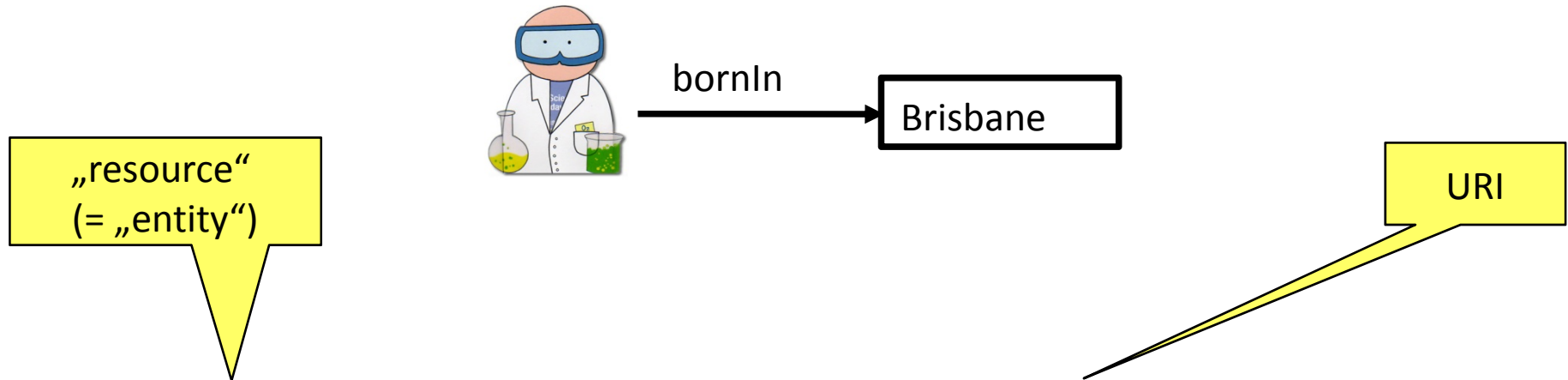


SamSmart livesIn Brisbane in 2009

living42	aboutPerson	SamSmart
living42	aboutPlace	Brisbane
living42	aboutTime	2009

Uniform Resource Identifiers (URIs)

A URI is similar to a URL, but it is not necessarily downloadable.
It identifies a concept uniquely.



SamSmart: <http://brisbane-corp.au/people/SamSmart>

bornIn: <http://mpii.de/yago/resource/bornIn>

Brisbane: <http://brisbane.au>

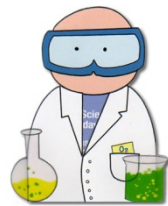
<http://www.ietf.org/rfc/rfc3986.txt>

URIs are used as globally unique identifiers for resources.

=> Knowledge can be interlinked. A knowledge base on one server can refer to concepts from another knowledge base on another server.

Namespaces

A namespace is a shorthand notation for the first part of a URI.



bornIn

Brisbane

Without namespaces,
our statement is
a triple of 3 URIs
-- quite verbose

<http://bsco.au/people/SamSmart> <http://mpii.de/yago/bornIn> <http://brisbane.au>

Namespace bsco := http://bsco.au/people/...

Namespace yago := http://mpii.de/yago/...

Namespaces make
our statement much
less verbose

bsco:SamSmart yago:bornIn <http://brisbane.au>

Namespaces are used to abbreviate URIs

=> Namespaces with useful concepts can become popular.

This facilitates a common vocabulary across different knowledge bases.

Popular Namespaces: Basic

- rdf: The basic RDF vocabulary
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- rdfs: RDF Schema vocabulary (predicates for classes etc., later in this talk)
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- owl: Web Ontology Language (for reasoning, later in this talk)
<http://www.w3.org/2002/07/owl#>
- dc: Dublin Core (predicates for describing documents, such as „author“, „title“ etc.)
<http://purl.org/dc/elements/1.1/>
- xsd: XML Schema (definition of basic datatypes)
<http://www.w3.org/2001/XMLSchema#>

Standard namespaces are used for basic concepts

=> The basic concepts are the same across all RDF knowledge bases

Popular Namespaces: Specific

dbp: The DBpedia ontology (real-world predicates and resources, e.g. Albert Einstein)
<http://dbpedia.org/resource/>

yago: The YAGO ontology (real-world predicates and resources, e.g. Albert Einstein)
<http://mpii.de/yago/resource/>

foaf: Friend Of A Friend (predicates for relationships between people)
<http://xmlns.com/foaf/0.1/>

cc: Creative Commons (types of licences)
<http://creativecommons.org/ns#>

.... and many, many more

There exist already a number of specific namespaces
=> Knowledge engineers don't have to start from scratch

Literals



example:SamSmart yago:bornIn <http://brisbane.au>

example:SamSmart rdfs:label „Sam Smart“^^xsd:string

We are using standard
RDF vocabulary here

The objects of statements
can also be literals

The literals can be typed.
Types are identified by a URI

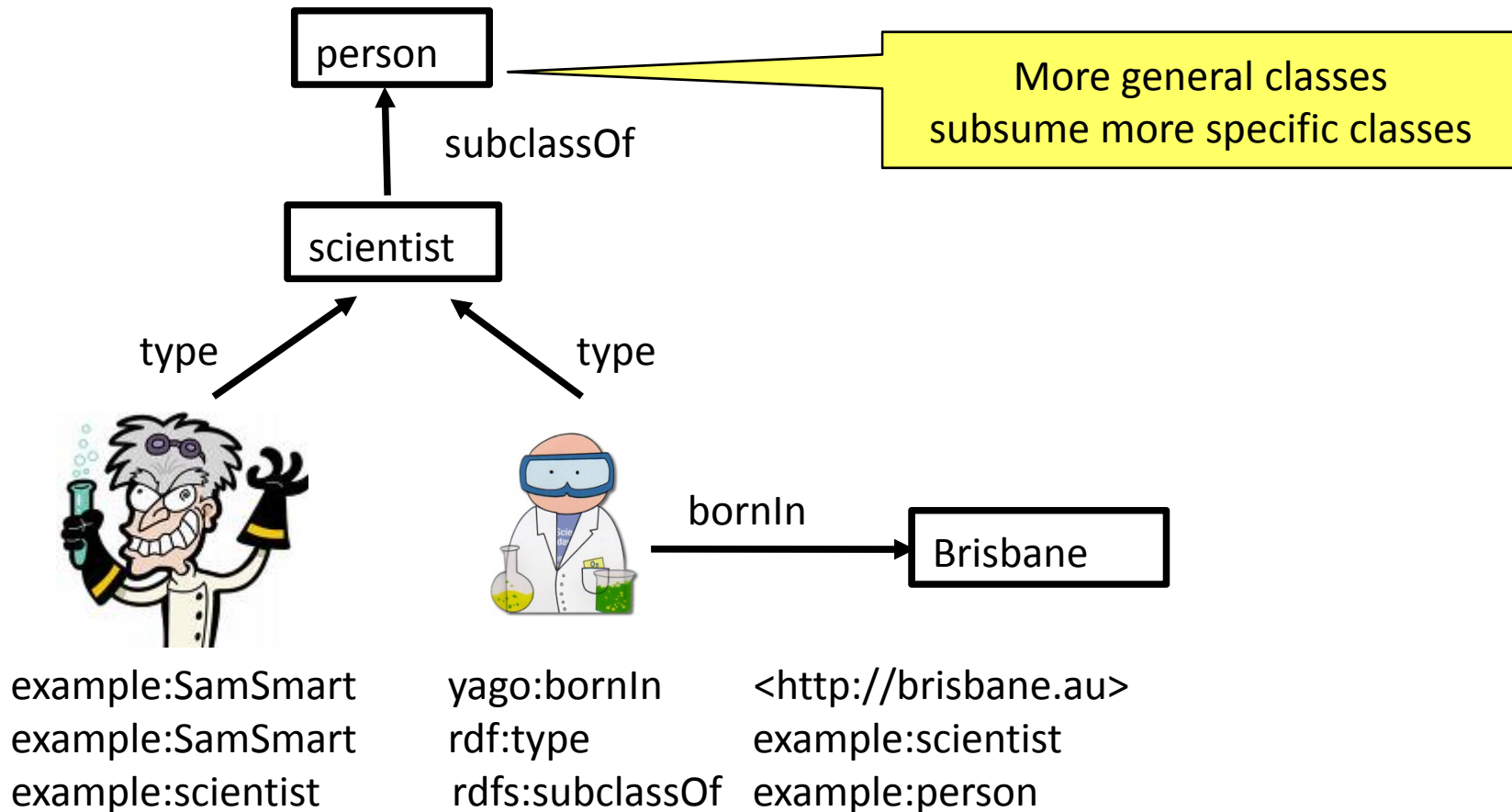
Popular types: xsd:string xsd:date xsd:nonNegativeInteger xsd:byte

Literals are can be labeled with pre-defined types
=> They come with a well-defined semantics.

<http://www.w3.org/TR/xmlschema-2/>₃₄

Classes

A class is a resource that represents a set of similar resources

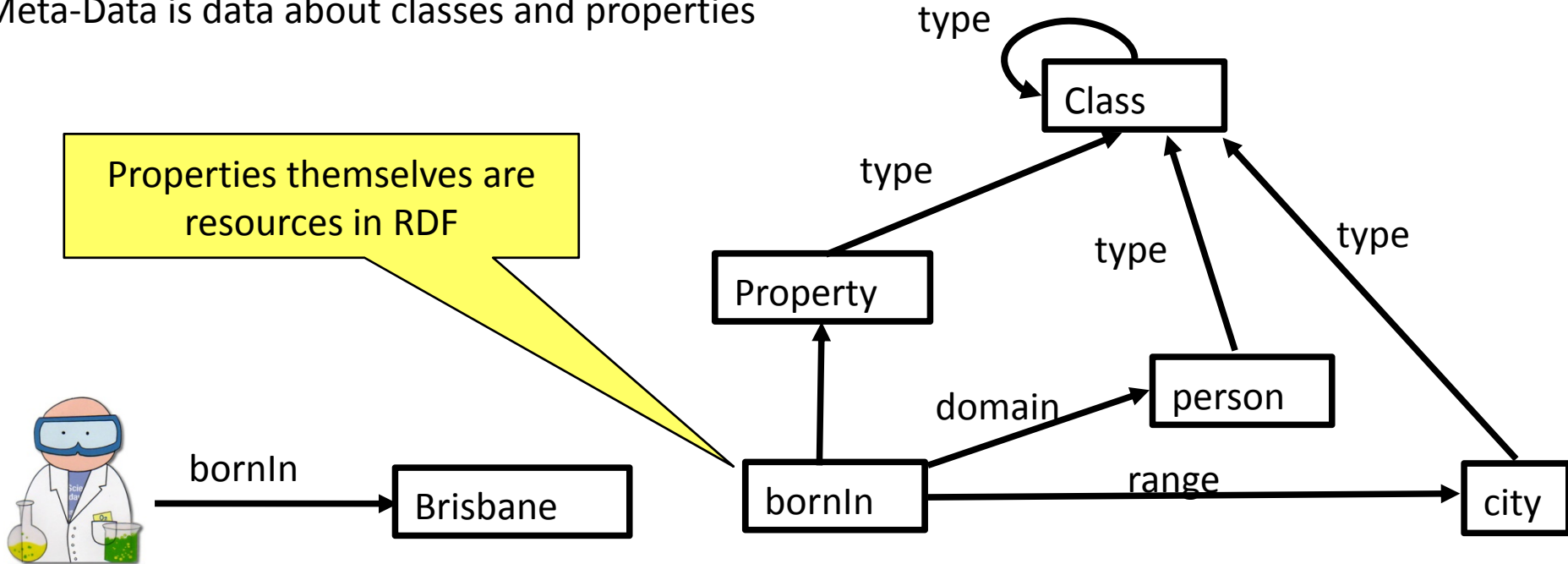


Due to historical reasons,
some vocabulary is
defined in RDF, other in RDFS

<http://www.w3.org/TR/rdf-schema/>

„Meta-Data“

Meta-Data is data about classes and properties



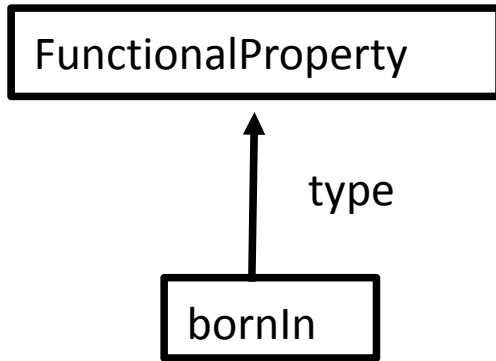
yago:bornIn	rdf:type	rdf:Property
yago:bornIn	rdfs:domain	example:person
yago:bornIn	rdfs:range	example:city
example:person	rdf:type	rdfs:Class
rdfs:Class	rdf:type	rdfs:Class

<http://www.w3.org/TR/rdf-schema/>

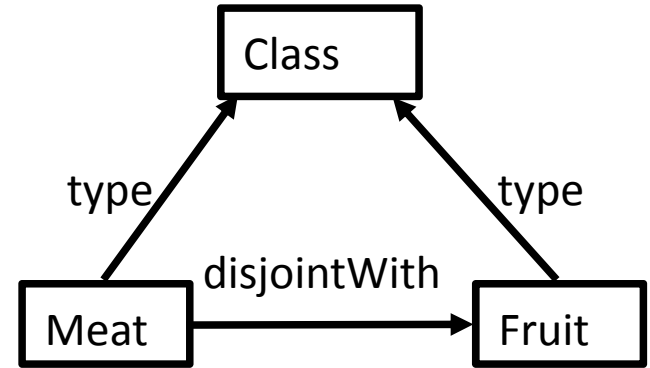
RDFS can be used to talk about classes and properties, too
=> There is no concept of „meta-data“ in RDFS

Reasoning

„A person can only be born in one place“



„Meat is not Fruit“



yago:bornIn
example:Meat

rdf:type
owl:disjointWith

owl:FunctionalProperty
example:Fruit

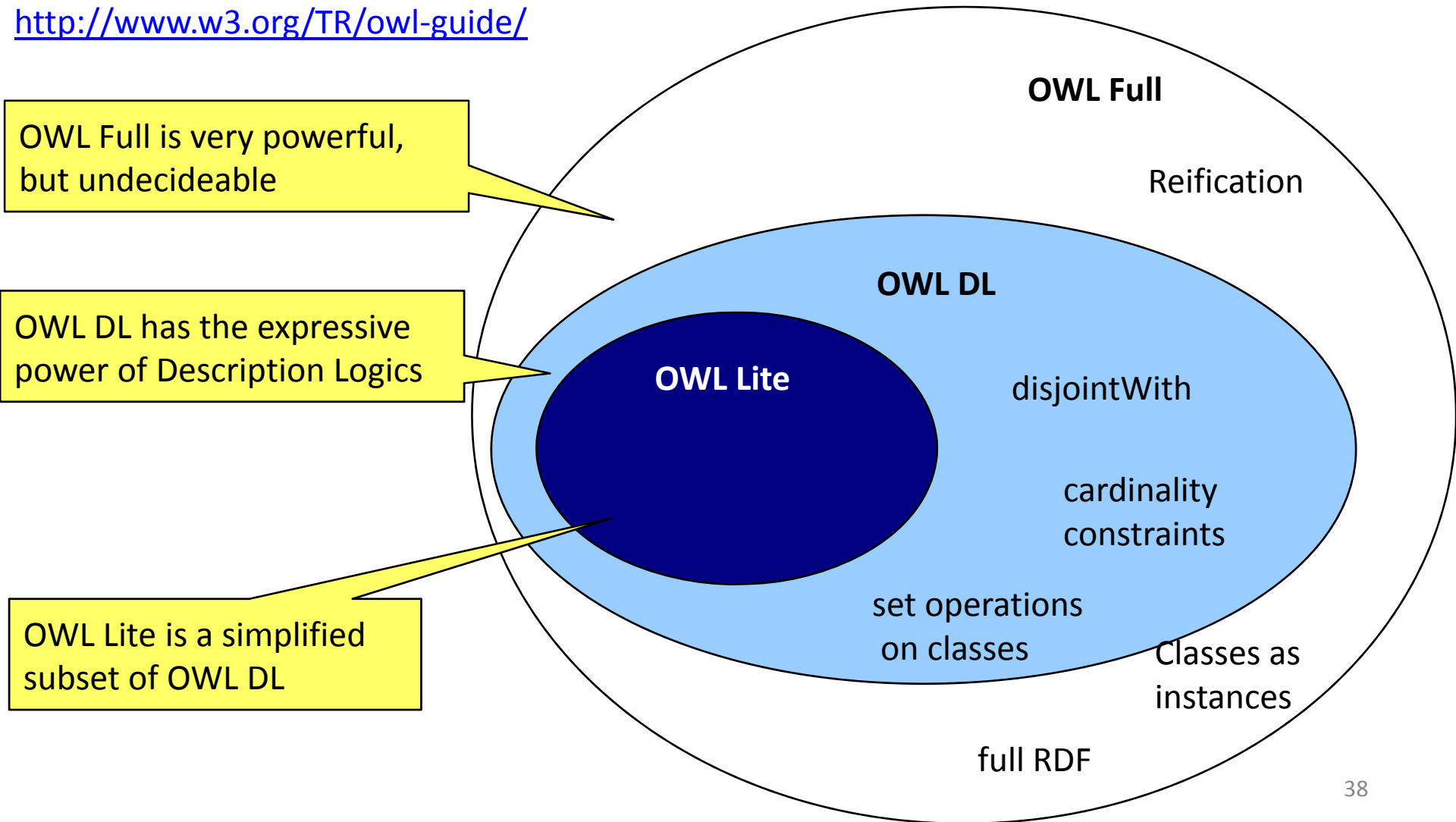
The owl namespace defines vocabulary for set operations on classes, restrictions on properties and equivalence of classes.

The OWL vocabulary can be used to express properties of classes and predicates
=> We can express logical consistency

Reasoning: Flavors of OWL

There exist 3 different flavors of OWL that trade off expressivity with tractability.

<http://www.w3.org/TR/owl-guide/>



Formats of RDF data

RDF is just the model of knowledge representation, there exist different formats to store it.

1. In a database („triple store“) with the schema

FACT(resource, predicate, resource)

2. As triples in plain text („Notation 3“, „Turtle“)

```
@prefix yago http://mpii.de/yago/resource
yago:SamSmart    yago:bornIn    <http://brisbane.au>
```

3. In XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:yago="http://mpii.de/yago/resource">
  <rdf:Description rdf:about="http://mpii.de/yago/resource/SamSmart">
    <yago:bornIn rdf:resource="http://brisbane.au" />
  </rdf:Description>
</rdf:RDF>
```

Existing OWL/RDF knowlegde bases: General

There exist already a number of knowledge bases in RDF.

Dataset	URL	#Statements
Freebase (community collaboration)	http://www.freebase.com	2.5m
OpenCyc (spin-off from commerical ontology Cyc)	http://www.opencyc.org	60k
DBpedia (extraction from Wikipedia, focus on coverage)	http://www.dbpedia.org	270m
YAGO (extraction from Wikipedia, focus on accuracy)	http://mpii.de/yago	20m

Existing OWL/RDF knowlegde bases: Specific

Dataset	URL	#Statements
MusicBrainz (Artists, Songs, Albums)	http://www.musicbrainz.org	23k
Geonames (Countries, Cities, Capitals)	http://www.geonames.org	85k
DBLP (Papers, Authors, Citations)	http://www4.wiwiss.fu-berlin.de/dblp/	15m
US Census (Population statistics)	http://www.rdfabout.com/demo/census/	1000m
...and many more....		

=> The Semantic Web has already a reasonable number of knowledge bases

Querying the knowledge bases: SPARQL

SPARQL is a query language for RDF data. It is similar to SQL

Which scientists are from Brisbane?

```
PREFIX rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#  
PREFIX example:....  
  
SELECT ?x WHERE {  
  ?x  rdf:type          example:scientist .  
  ?x  example:bornIn   example:Brisbane  
}
```

Define our namespaces

Pose the query in SQL style

<http://www.w3.org/TR/rdf-sparql-query/>

Sample Query on YAGO

Which scientists are from Brisbane?

Yago - A Core of Semantic Knowledge - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://www.mpi-inf.mpg.de/yago-naga/yago/

Home
Use YAGO
Query YAGO
References
Related Projects
Acknowledgements

Query Form

YAGO-query:

?id0:	?x	type	scientist
?id1:	?x	bornIn	Brisbane
?id2:			
?id3:			

Daten absenden

?Brisbane = [Brisbane](#)
?scientist = [scientist](#)
?x = [R. J. McKay](#)

?Brisbane = [Brisbane](#)
?scientist = [scientist](#)
?x = [Peter C. Doherty](#)

Fertig

References

Specifications

RDF	http://www.w3.org/TR/rdf-primer/
RDFS	http://www.w3.org/TR/rdf-schema/
URIs	http://www.ietf.org/rfc/rfc3986.txt
Literals	http://www.ietf.org/rfc/rfc3986.txt
OWL	http://www.w3.org/TR/owl-guide/
SPARQL	http://www.w3.org/TR/rdf-sparql-query/

Projects

YAGO	Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum „YAGO - A Core of Sematic Knowledge“ (WWW 2007)
DBpedia	S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, Z. Ives „DBpedia: A Nucleus for a Web of Open Data“ (ISWC 2007)
LOD	Christian Bizer, Tom Heath, Danny Ayers, Yves Raimond „Interlinking Open Data on the Web“ (ESWC 2007)

Section 3: XML Mining

Richi Nayak

Faculty of Information Technology
Queensland University of Technology
Brisbane, Australia

r.nayak@qut.edu.au

Outline

- What XML is?
- What XML Mining is?
- Why should we do XML mining?
- How we do XML mining?
- Future directions

XML

- XML: eXtensible Markup Language
- XML v. HTML
 - HTML: restricted set of tags, e.g. <TABLE>, <H1>, , etc.
 - XML: you can create your own tags
- Selena Sol (2000) highlights the four major benefits of using XML language:
 - XML separates data from presentation which means making changes to the display of data does not affect the XML data;
 - Searching for data in XML documents becomes easier as search engines can parse the description-bearing tags of the XML documents;
 - XML tag is human readable, even a person with no knowledge of XML language can still read an XML document;
 - Complex structures and relations of data can be encoded using XML.

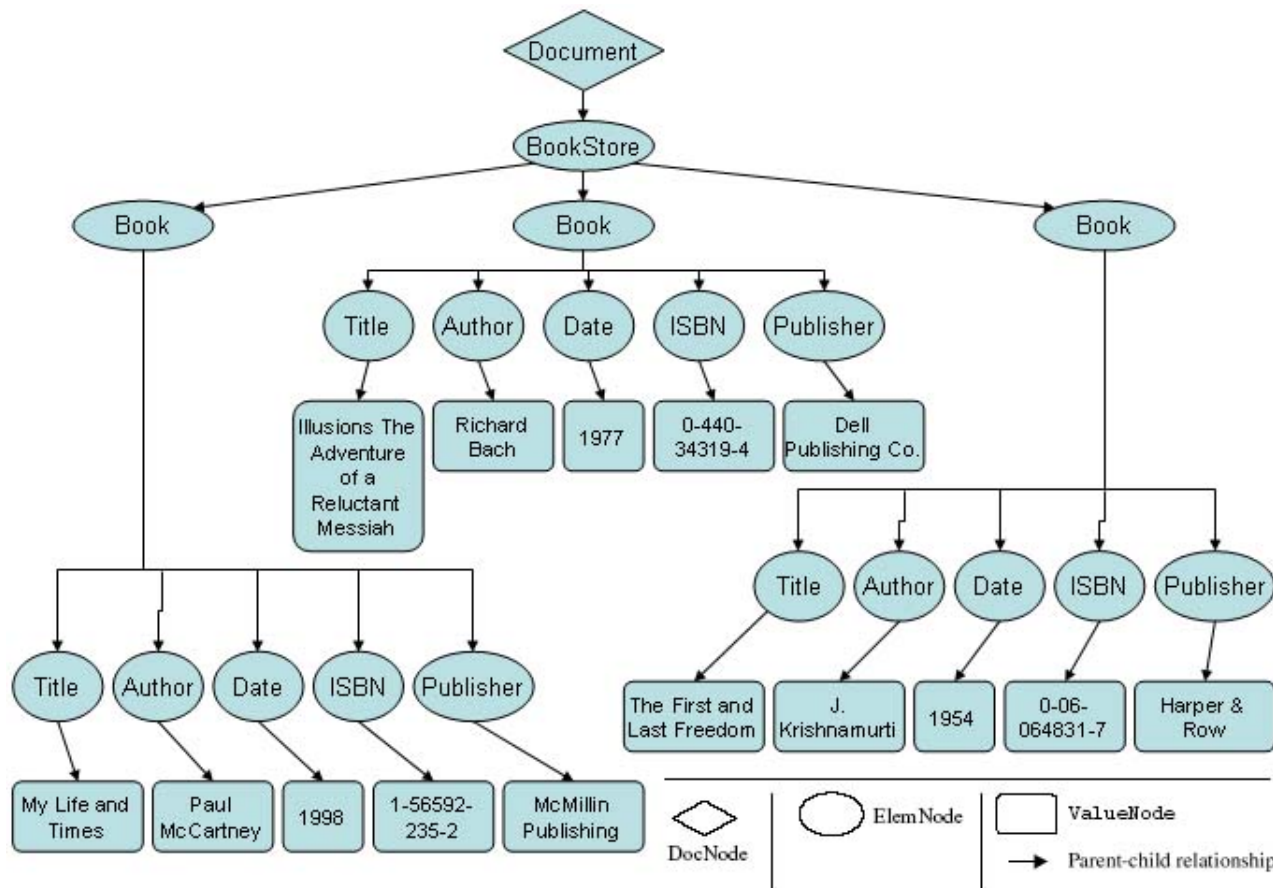
XML: An Example

- XML is a semi structured language

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tom</to>
  <from>Mary</from>
  <heading>Reminder</heading>
  <body>
    Tomorrow is meeting.
  </body>
</note>
```

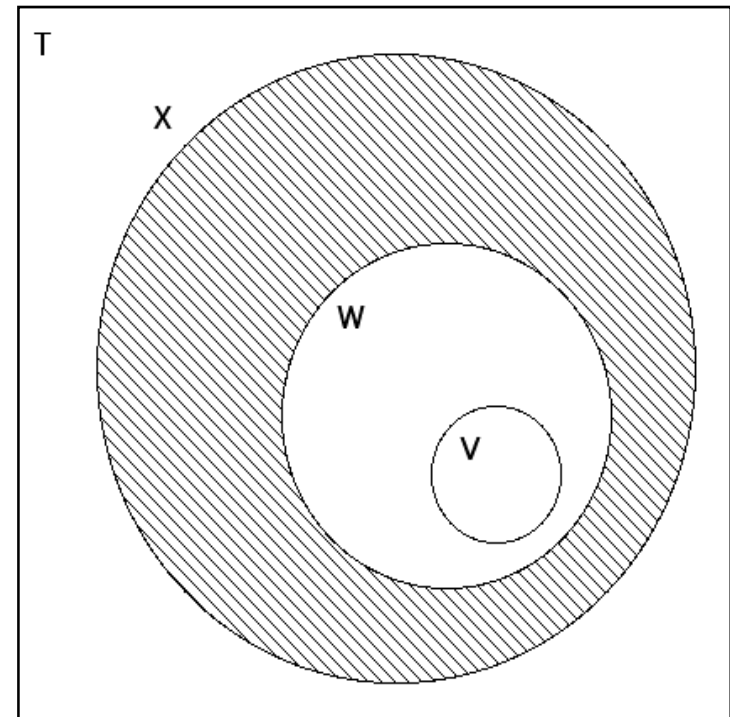
XML: Data Model

XML can be represented as a tree or graph oriented data model.



XML Schemas

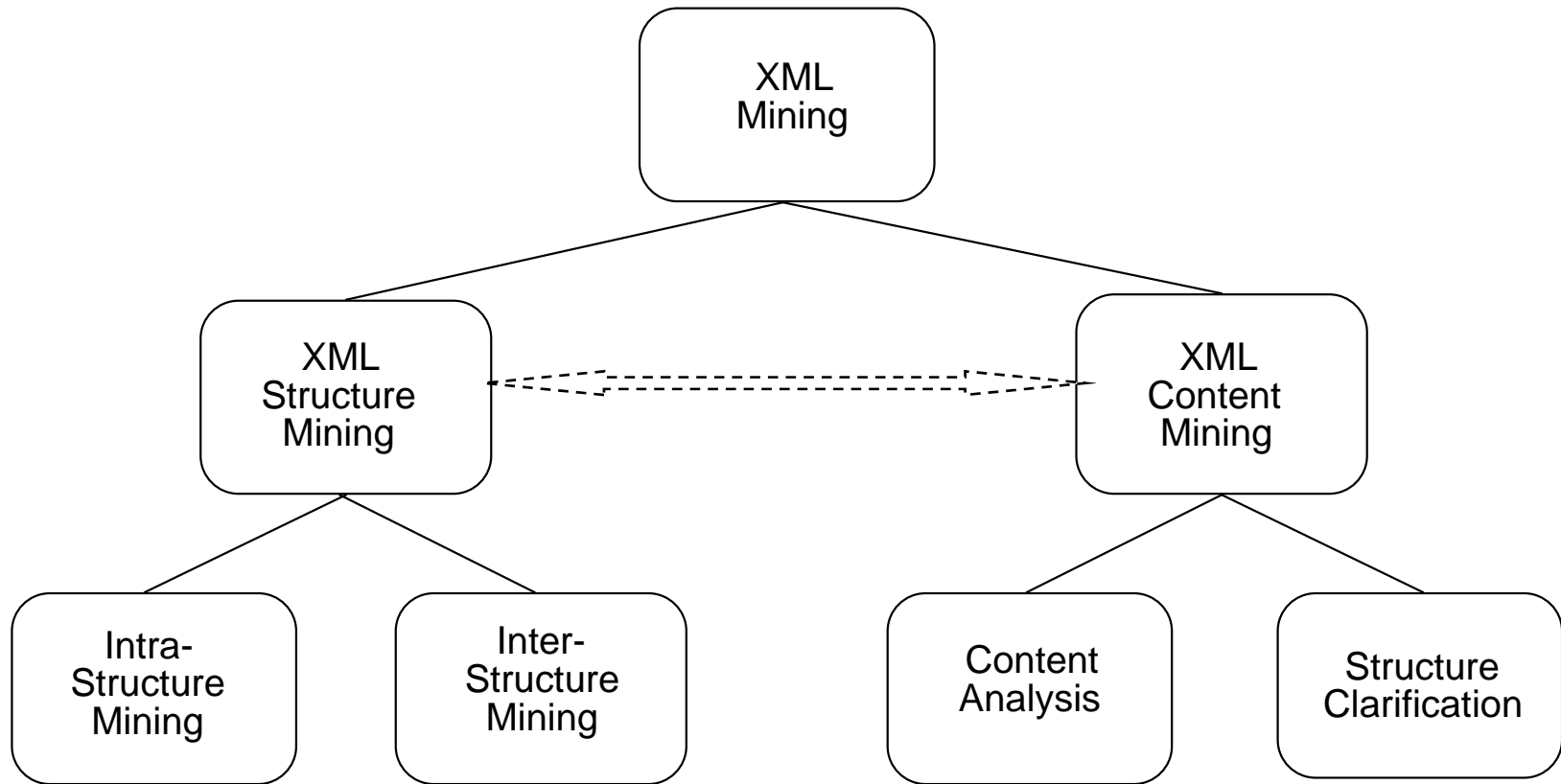
- XML allows the possibility of defining document schema.
- Document schema contains the grammar for restricting syntax and structure of XML documents.
- Two commonly used schemas are:
 - Document Type Definition (DTD)
 - XML Schema Definition (XSD)
 - Allows more extensive data-checking
- Valid XML documents conforms to its schema.



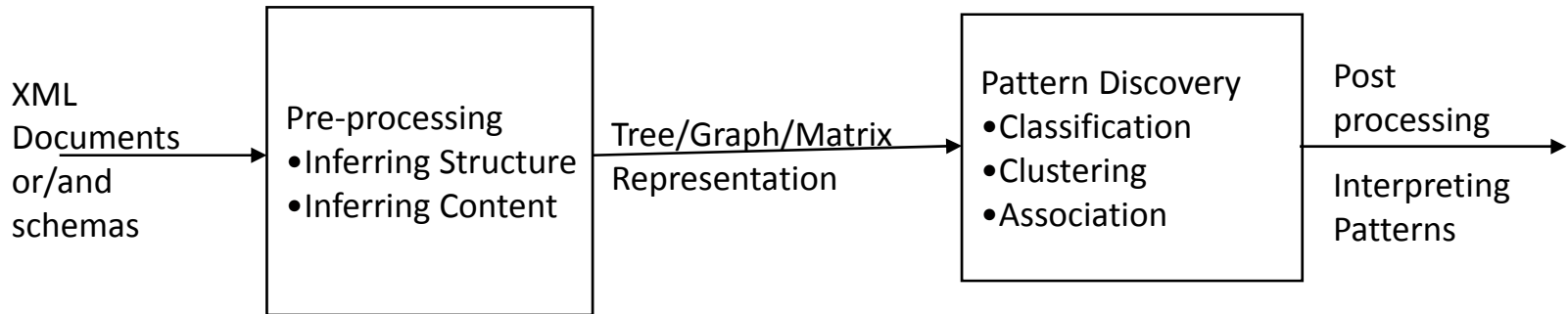
Requirements for XML mining

- What is specific to XML data that defines the requirements for XML mining?
 - Structures and Content
 - Flexibility in its design
 - Multimodal
 - Scalability
 - Heterogeneous
 - Online
 - Distributed
 - Autonomous

A XML Mining Taxonomy



XML Mining Process



d₁

<R>

<E₁>t₁, t₂, t₃
 <E₂>t₄, t₃, t₆
 <E₃>t₅, t₄, t₇
 <E_{3.1}>t₅, t₂, t₁
 <E_{3.2}>t₇, t₉

d₃

<R>

<E₁>t₁, t₂
 <E₂>t₃, t₃
 <E₃>t₅, t₄, t₇
 <E_{3.1}>t₅, t₂, t₁
 <E_{3.2}>t₇, t₉

d₂

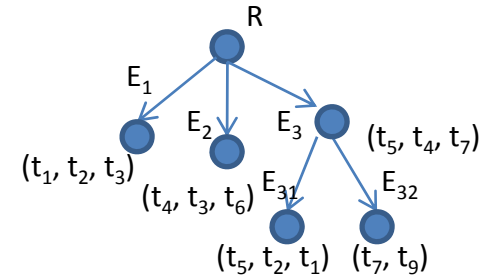
<R>

<E₁>t₁, t₄
 <E₂>t₃, t₃
 <E₃>t₄, t₇
 <E_{3.1}>t₂, t₉
 <E_{3.2}>t₂, t₇, t₈, t₁₀

d₄

<R>

<E₁>t₁, t₄
 <E₃>t₄, t₇
 <E₃>t₄, t₈
 <E₁>t₁, t₄



Equivalent Tree Representation

Four Example XML Documents

	d ₁	d ₂	d ₃	d ₄
t ₁	2	1	2	2
t ₂	2	2	2	0
t ₃	2	2	2	0
t ₄	2	2	1	4
t ₅	2	0	2	0
t ₆	1	0	0	0
t ₇	2	2	2	1
t ₈	0	1	0	1
t ₉	1	1	1	0
t ₁₀	0	1	0	0

Equivalent Content Matrix Representation

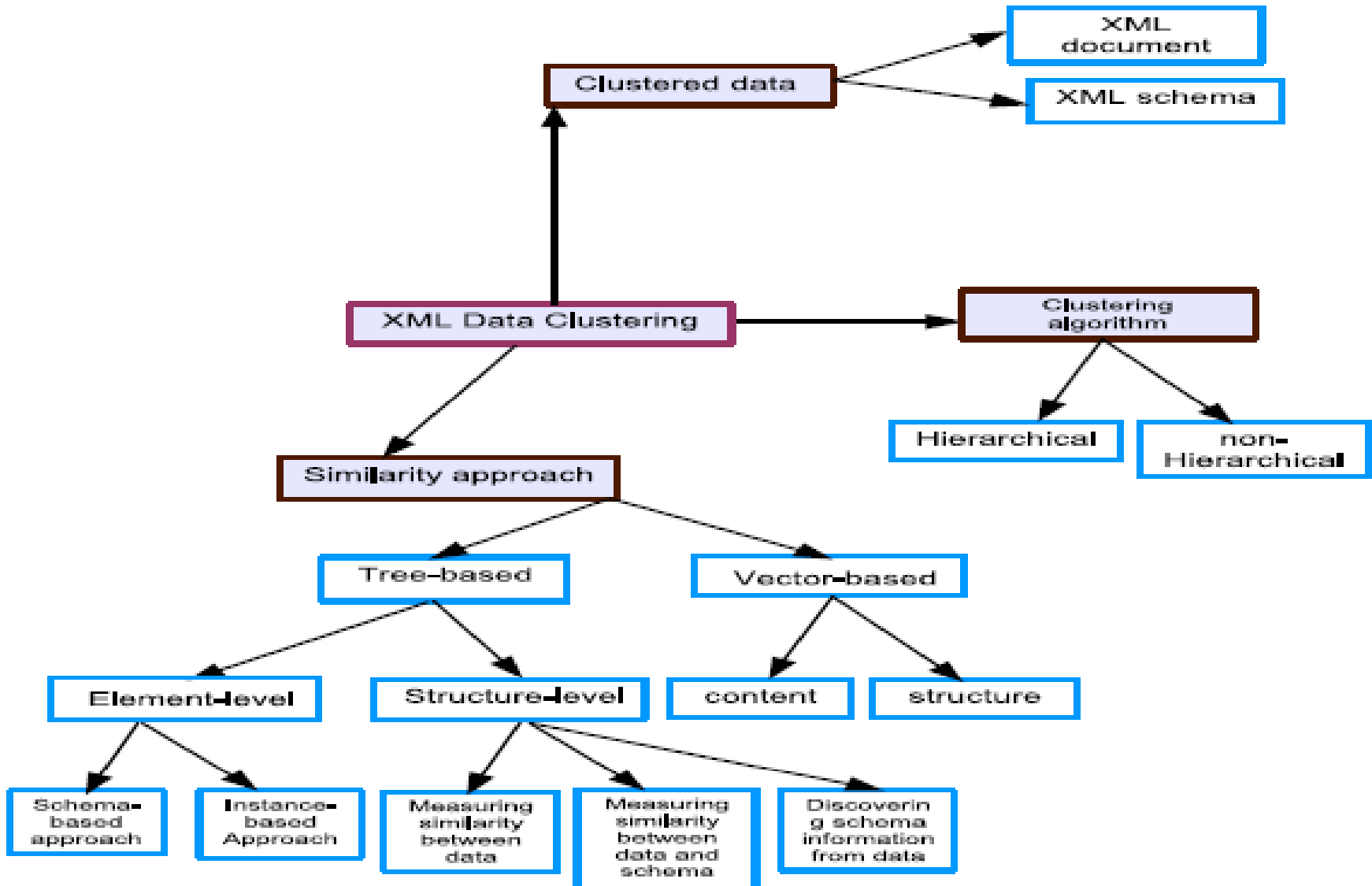
	d ₁	d ₂	d ₃	d ₄
R/E ₁	1	1	1	2
R/E ₂	1	1	1	0
R/E ₃ /	1	2	1	0
E _{3.1}				
R/E ₃ /	1	0	1	0
E _{3.2}				
R/E ₃	1	1	1	2

Equivalent Structure Matrix Representation

Some Mining Examples

- Mining frequent tree patterns
- Grouping and classifying documents/schemas
- Schema discovery
- Schema-based mining
- Mining association rules
- Mining XML queries
- Etc.

XML Clustering: Types and Approaches



XML Clustering: Data Models and Methods

- Structure
 - Edit distance (string, tree, ordered tree, graph)
 - Vector Space Models
- Content
 - Vector Space Models
- Mixing Structure and Content
 - Vector Space Models
 - Tensor models

The clustering process

- Find similarities between XML sources
 - by considering the XML semantic information such as the linguistic and the context of the elements
 - as well as the hierarchical structure information such as parent, children, and siblings.
- The process usually starts by considering the tree structures, as derived in the pre-processing step.
- The semantic similarity is measured by comparing each pair of elements of two trees primarily based on their names taking into account the acronyms, synonyms, hyponyms, hypernyms.
- The structural similarity is measured by considering the hierarchical positions of elements in the tree.
 - The utilization of sequential patterns mining algorithms has been used by many researchers to measure structural similarity.
- The semantic and structural similarity is combined to measure how similar two documents are.
- The pair-wise matrix becomes input for a clustering algorithm.

Frequent Tree Mining

- XML sources are generally represented as an ordered labelled or unordered labelled tree.
- The task is to build up associations among trees (or sub-trees or sub-graphs or paths) rather than items as in traditional mining.
- The frequent tree mining extracts substructures that occur frequently among a set of XML documents or within an individual XML document.
- These frequent substructures generate association rules.
- However, the frequent substructures are hierarchical and counting support requires more than just the join of flat sets.

Classifications of Tree Mining algorithms

Based on:

- Tree Representation
 - Free trees, Rooted Unordered Tree, Rooted Ordered Tree
- Subtree Representation
 - Induced Subtree, Embedded Subtree
- Traversal strategy
 - Depth-first, Breadth-first, Depth-first & Breadth-first

Classifications of Tree Mining algorithms

Based on:

- Canonical representation
 - Pre-order string encoding, Level-wise encoding
- Tree mining approach
 - Candidate generation (extension, Join), Pattern-growth
- Condensed representation
 - Closed, Maximal

XML Classification Mining

- The task is to find structural rules in order to classify XML documents into the set of predefined classifications of documents.
- In the training phase, a set of structural classification rules are built that can be used in the learning phase to classify data (with unknown classes).
- The existing classification algorithms are not efficient to classify the XML documents because they are not capable of exploring the structural information.
- Few researchers have developed generic (e.g., information retrieval (IR) based and association based) classifiers as well as specific (e.g. rule based according to structures) classifiers for XML.

XML Classification Mining

- The IR-based methods treat each document as a “bag of words”.
 - These methods use the actual text of the XML data, and do not take into account a considerable amount of structural information inside the documents.
- The association-based methods use the associations among different nodes visited in a session in order to perform the classification.
- An effective rule-based classifier for XML, XRules, uses a set of structural rules for the classification of XML documents.
 - It first mines frequent structures in a collection of XML trees.
 - The frequent structures according to their support count for each class of documents are generated.
 - The next task is to find distinction between groups of rules for each class so a group of rules can uniquely define a class.
 - XRules uses the *bayesian induction algorithm* to combine the strength of structure frequency and an optimal neighbourhood ratio for a given set of documents.

Future Directions

- Scalability
 - Incremental Approaches
- Combining structure and content efficiently
 - Advanced data representational models and mining methods
- Application Context

Summary

- XML mining, in order to be more than a temporary fade, must deliver useful solutions for practical applications.
- Applications with large amounts of raw strategic data in XML will be there.
- XML data mining techniques will be a plus for the adoption of XML as a data model for modern applications.

Reading Articles

- R. Nayak (2008) "XML Data Mining: Process and Applications", Chapter 15 in "Handbook of Research on Text and Web Mining Technologies", Ed: Min Song and Yi-Fang Wu. Publisher: Idea Group Inc., USA. PP. 249-271.
- S. Kutty and R. Nayak (2008) "Frequent Pattern Mining on XML documents", Chapter 14 in "Handbook of Research on Text and Web Mining Technologies", Ed: Min Song and Yi-Fang Wu. Publisher: Idea Group Inc., USA. PP. 227-248.
- R. Nayak (2008) "Fast and Effective Clustering of XML Data Utilizing their Structural Information". Knowledge and Information Systems (KAIS). Volume 14, No. 2, February 2008 pp 197-215.
- C. C. Aggarwal, N. Ta, J. Wang, J. Feng, and M. Zaki, "Xproj: a framework for projected structural clustering of xml documents," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining San Jose, California, USA: ACM, 2007, pp. 46-55.
- Nayak, R., & Zaki, M. (Eds.). (2006). Knowledge Discovery from XML documents: PAKDD 2006 Workshop Proceedings (Vol. 3915): Springer-Verlag Heidelberg.
- NAYAK, R. AND TRAN, T. 2007. A progressive clustering algorithm to group the XML data by structural and semantic similarity. *International Journal of Pattern Recognition and Artificial Intelligence* 21, 4, 723-743.
- Y. Chi, S. Nijssen, R. R. Muntz, and J. N. Kok, "Frequent Subtree Mining- An Overview," in *Fundamenta Informaticae*. vol. 66: IOS Press, 2005, pp. 161-198.
- L. Denoyer and P. Gallinari, "Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents," SIGIR Forum, vol. 41, pp. 79-90, 2007.
- BERTINO, E., GUERRINI, G., AND MESITI, M. 2008. Measuring the structural similarity among XML documents and DTDs. *Intelligent Information Systems* 30, 1, 55-92.
- BEX, G. J., NEVEN, F., AND VANSUMMEREN, S. 2007. Inferring XML schema definitions from XML data. In Proceedings of the 33rd International Conference on Very Large Data Bases. Vienna, Austria, 998-1009.
- BILLE, P. 2005. A survey on tree edit distance and related problems. *Theoretical Computer Science* 337, 1-3, 217-239.
- BONIFATI, A., MECCA, G., PAPPALARDO, A., RAUNICH, S., AND SUMMA, G. 2008. Schema mapping verification: the spicy way. In *EDBT*. 85-96.

Related Publications

- BOUKOTTAYA, A. AND VANOIRBEEK, C. 2005. Schema matching for transforming structured documents. In *DocEng'05*. 101–110.
- FLESCA, S., MANCO, G., MASCIARI, E., PONTIERI, L., AND PUGLIESE, A. 2005. Fast detection of XML structural similarity. *IEEE Trans. on Knowledge and Data Engineering* 17, 2, 160–175.
- GOU, G. AND CHIRKOVA, R. 2007. Efficiently querying large XML data repositories: A survey. *IEEE Trans. on Knowledge and Data Engineering* 19, 10, 1381–1403.
- NAYAK, R. AND IRYADI, W. 2007. XML schema clustering with semantic and hierarchical similarity measures. *Knowledge-based Systems* 20, 336–349.
- Kutty, S., Nayak, R., & Li, Y. (2007). *PCITMiner- Prefix-based Closed Induced Tree Miner for finding closed induced frequent subtrees*. Paper presented at the the Sixth Australasian Data Mining Conference (AusDM 2007), Gold Coast, Australia.
- TAGARELLI, A. AND GRECO, S. 2006. Toward semantic XML clustering. In *SDM 2006*. 188–199.
- Rusu, L. I., Rahayu, W., & Taniar, D. (2007). Mining Association Rules from XML Documents. In A. Vakali & G. Pallis (Eds.), *Web Data Management Practices*:
- Li, H.-F., Shan, M.-K., & Lee, S.-Y. (2006). Online mining of frequent query trees over XML data streams. In *Proceedings of the 15th international conference on World Wide Web* (pp. 959-960). Edinburgh, Scotland: ACM Press.
- Zaki, M. J.:(2005):Efficiently mining frequent trees in a forest: algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 17 (8): 1021-1035
- Zaki, M. J., & Aggarwal, C. C. (2003). *XRULES: An Effective Structural Classifier for XML Data*. Paper presented at the SIGKDD.
- Wan, J. W. W. D., G. (2004). Mining Association rules from XML data mining query. *Research and practice in Information Technology*, 32, 169-174.

Section 4: Domain-Specific Markup Languages

Aparna Varde

Department of Computer Science

Montclair State University

Montclair, NJ, USA

vardea@mail.montclair.edu

What is a Domain-Specific Markup Language?

- Medium of communication for users of the domain
- Follows XML syntax
- Encompasses the semantics of the domain



Examples of Domain-Specific Markup Languages

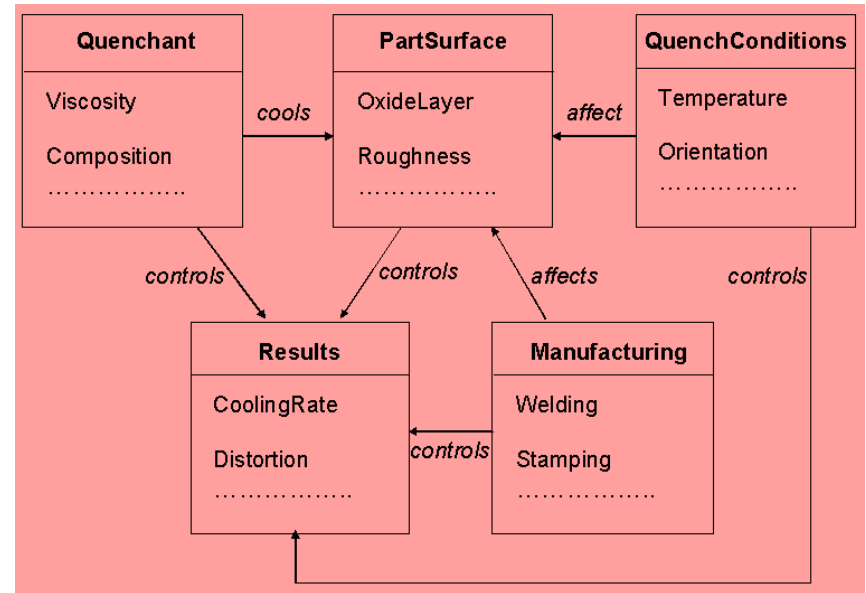
- MML: Medical Markup Language
- ChemML: Chemical Markup Language
- MatML: Materials Markup Language
- AniML: Analytical Information Markup Language
- MathML: Mathematics Markup Language
- WML: Wireless Markup Language

Steps in Markup Language Development

1. Domain Knowledge Acquisition
2. Ontology Creation
3. Schema Development

Domain Knowledge Acquisition

- Terminology Study
 - Understand concepts in domain well
 - Find out if new markup language should be an extension to an existing markup or an independent language
- Data Modeling
 - Use ER models, UML etc.
 - This also serves as a medium of communication
- Requirements Specifications
 - Conduct interviews with domain experts who can convey user needs
 - Develop Requirement Specifications accordingly



Example of ER model for Heat Treating of Materials in Materials Science domain

Ontology Creation

- Ontology is a system of nomenclature used in a given domain
- Important considerations in ontology are synonyms and homographs
- Once initial ontology is established, it is useful to have discussions with experts and other users to make changes
- Revision of the ontology can go through several rounds of discussion and testing


- **Quenchant:** This refers to the medium used for cooling in the heat treatment process of rapid cooling or Quenching.
– *Alternative Term(s): CoolingMedium*
- **PartSurface:** The characteristics pertaining to the surface of the part undergoing heat treatment are recorded here.
– *Alternative Term(s): ProbeSurface, WorkpieceSurface*
- **Manufacturing:** The details of the processes used in the production of the concerned part such as welding and stamping are stored here.
– *Alternative Term(s): Production*
- **QuenchConditions:** This records the input parameters under which the Quenching process occurs, e.g., the temperature of the cooling medium, the extent to which the medium is agitated and so forth.
– *Alternative Term(s): InputConditions, InputParameters, QuenchParameters*
- **Results:** This stores the outcome of the Quenching process in terms of properties such as cooling rate (change in part temperature with respect to time) and heat transfer coefficient (measurement of heat extraction capacity of the whole process of rapid cooling).
– *Alternative Term(s): Output, Outcome*

Example of Ontology for QuenchML:
Quenching Markup Language for
Heat Treating of Materials

Schema Development

- Schema provides the structure of the markup language
- E-R model, requirements specification and ontology serve as the basis for schema design
- Each entity in E-R model significant in requirements specification typically corresponds to a schema element
- First schema draft is revised until users are satisfied that it adequately represents their needs
- Schema revision may involve several iterations, including discussions with standards bodies

```
<Quenching>
  <Quenchant>
  </Quenchant>
  <PartSurface>
  </PartSurface>
  <Manufacturing>
  </Manufacturing>
  <QuenchConditions>
  </QuenchConditions>
  <Results>
  </Results>
  <Graphs>
  </Graphs>
</Quenching>
```



```
<Results>
  <CoolingRate>
    <CRLocation>
      <CRValue>
      </CRValue>
    </CRLocation>
  </CoolingRate>
  <CoolingUniformity>
  </CoolingUniformity>
  <HeatTransferCoefficient>
    <Surface>
      <HCValue>
      </HCValue>
    </Surface>
  </HeatTransferCoefficient>
  <Hardness>
  </Hardness>
  <Distortion>
  </Distortion>
  <QuenchSeverity>
  </QuenchSeverity>
</Results>
```

Example Partial Snapshot of
QuenchML Schema

Desired Properties of Markup Languages

- **Avoidance of Redundancy**
 - If information about an entity or attribute is stored in an existing markup language, it should not be repeated in the new markup language
 - E.g., Thermal Conductivity stored in MatML, do not repeat in QuenchML
- **Non-Ambiguous Presentation of Information**
 - Consider concepts such as synonyms, e.g., in Salary and Income, and homographs, e.g., Share (part of something or stocks) in Financial fields
- **Easy Interpretability of Information**
 - Readers should be able to understand stored information without much reference to related documentation
 - E.g., in Scientific fields, store Input Conditions of experiments before Results
- **Incorporation of Domain-Specific Requirements**
 - Issues such as primary keys, e.g., Student ID in Academic fields

Application of XML Features in Language Development

1. Sequence Constraint
2. Choice Constraint
3. Key Constraint
4. Occurrence Constraint

Sequence Constraint

```
<xsd:element name="Quenching">
  <xsd:complexType>
    <xsd:sequence>
      .....
      <xsd:element name="QuenchConditions">
        .....
      </xsd:element>
      <xsd:element name="Results"/>
        .....
      </xsd:element>
      .....
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

- Used to declare elements to occur in a certain order
- Example:
 - Quenching is a step in Heat Treatment of Materials
 - QuenchML proposed as extension to MatML
 - *QuenchConditions* must come before *Results* for meaningful interpretation

Choice Constraint

```
<xsd:element name="Manufacturing">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="Casting"/>
      <xsd:element ref="PowderMetallurgy"/>
    </xsd:choice>
    .....
  </xsd:complexType>
</xsd:element>
```

- Used to declare mutually exclusive elements, i.e., only one of them can exist
- Example
 - In Heat Treating, part being heated can be manufactured by either *Casting* or *Powder Metallurgy*, not both
 - In Finance, a person can be either *Solvent* or *Bankrupt*, not both

Key Constraint

```
<xsd:element name="Quenchant">
  <xsd:complexType>
    <xsd:attribute name="id" type="xsd:ID" use="required"/>

    .....
  </xsd:complexType>
</xsd:element>
```

- Used to declare an attribute to be a unique identifier
- Analogous to primary key in relational databases
- Example:
 - In Heat Treating, name of Quenchant
 - In Census Applications, SSN of a person

Occurrence Constraint

```
<xsd:element name="Cooling Rate" minOccurs="8"
maxOccurs="unbounded">
.....
</xsd:element>

<xsd:element name="Graphs" minOccurs="0"
maxOccurs="3">
.....
</xsd:element>
```

- Used to declare minimum and maximum permissible occurrences of an element
- Example:
 - In Heat Treating, Cooling Rate must be recorded for at least 8 points, no upper bound
 - In same context, at most 3 Graphs are stored, no lower bound

Convenient Access to Information for Knowledge Discovery

1. XQuery: XML Query Language
2. XSLT: XML Style Sheet Language Transformation
3. XPath: XML Path Language

XQuery

- XQuery (XML Query Language) developed by the World Wide Web Consortium (W3C)
- XQuery can retrieve information stored using domain-specific markup languages designed with XML tags
- It is thus advisable to design the markup language to facilitate retrieval using XQuery
 - Storing data in a case sensitive manner
 - Using additional tags for storage to enhance querying efficiency

XSLT

- XSLT stands for XML Style Sheet Language Transformations
- It is a language for transforming XML documents into other XML documents
- This includes an XML vocabulary for specifying formatting
- Information stored using an XML based Markup Language is easily accessible through XSLT

XPath

- XPath, the XML Path Language, is a language for addressing parts of an XML document
- In support of this primary purpose, it also provides basic facilities for manipulation of strings, numbers and booleans
- XPath models an XML document as a tree of nodes
- There are different types of nodes, including element nodes, attribute nodes and text nodes
- XPath fully supports XML Namespaces
- All this further enhances the retrieval of information with reference to context

Data Mining with Association Rules

- Association Rules are of the type $A \Rightarrow B$
 - Example: fever \Rightarrow flu
- Interestingness measures
 - Rule confidence : $P(B/A)$
 - Rule support: $P(A \cup B)$
- Data stored in a markup language facilitates rule derivation over text sources of information
- This helps to discover knowledge from text data

- `<fever> yes </fever>` in 9/10 instances
- `<flu> yes </flu>` in 7/10 instances
 - 6 of these in common with fever
- This helps to discover a rule
fever = yes \Rightarrow flu = yes
- Rule confidence: $6/9 = 67\%$
- Rule support: $6/10 = 60\%$

Real World Applications

- Data stored using markup languages can be used to develop efficient Management Information Systems (MIS) in given domains
- Rule derivation from text sources can serve as basis for knowledge discovery to develop Expert Systems
- Other techniques such as document clustering can be applied over text data stored using markup languages for better Information Retrieval

References

1. Boag, S., Fernandez, M., Florescu, D., Robie J. and Simeon, J.: XQuery 1.0: An XML Query Language, W3C Working Draft, November 2003.
2. Clark, J. and DeRose, S.: XML Path Language (XPath) Version 1.0. W3C Recommendation, Nov 1999.
3. Davidson, S., Fan, W., Hara, C. and Qin, J.: Propagating XML Constraints to Relations. In International Conference on Data Engineering, March 2003.
4. Guo, J., Araki, K., Tanaka, K., Sato, J., Suzuki, M., Takada, A., Suzuki, T., Nakashima, Y. and Yoshihara, H.: The Latest MML (Medical Markup Language) —XML based Standard for Medical Data Exchange / Storage. In: Journal of Medical Systems, Vol. 27, No. 4, pp. 357 – 366, Aug 2003.
5. Varde, A., Rundensteiner, E. and Fahrenholz, S.: XML Based Markup Languages for Specific Domains, Book Chapter, In Web Based Support Systems", Springer, 2008.

Conclusions

- Developments in Web technology outlined
 - Deep Web
 - Semantic Web
 - XML
 - Domain Specific Markup Languages
- Discussion on how these developments facilitate knowledge discovery included
- Suitable examples and applications provided